昭和58年6月7日国鉄首都特別扱承認雑誌第6895号　昭和58年8月1日発行(毎偶数月1日発行)第1巻第4号

FUJITSU PERSONAL COMPUTER MAGAZINE
パソコン情報誌

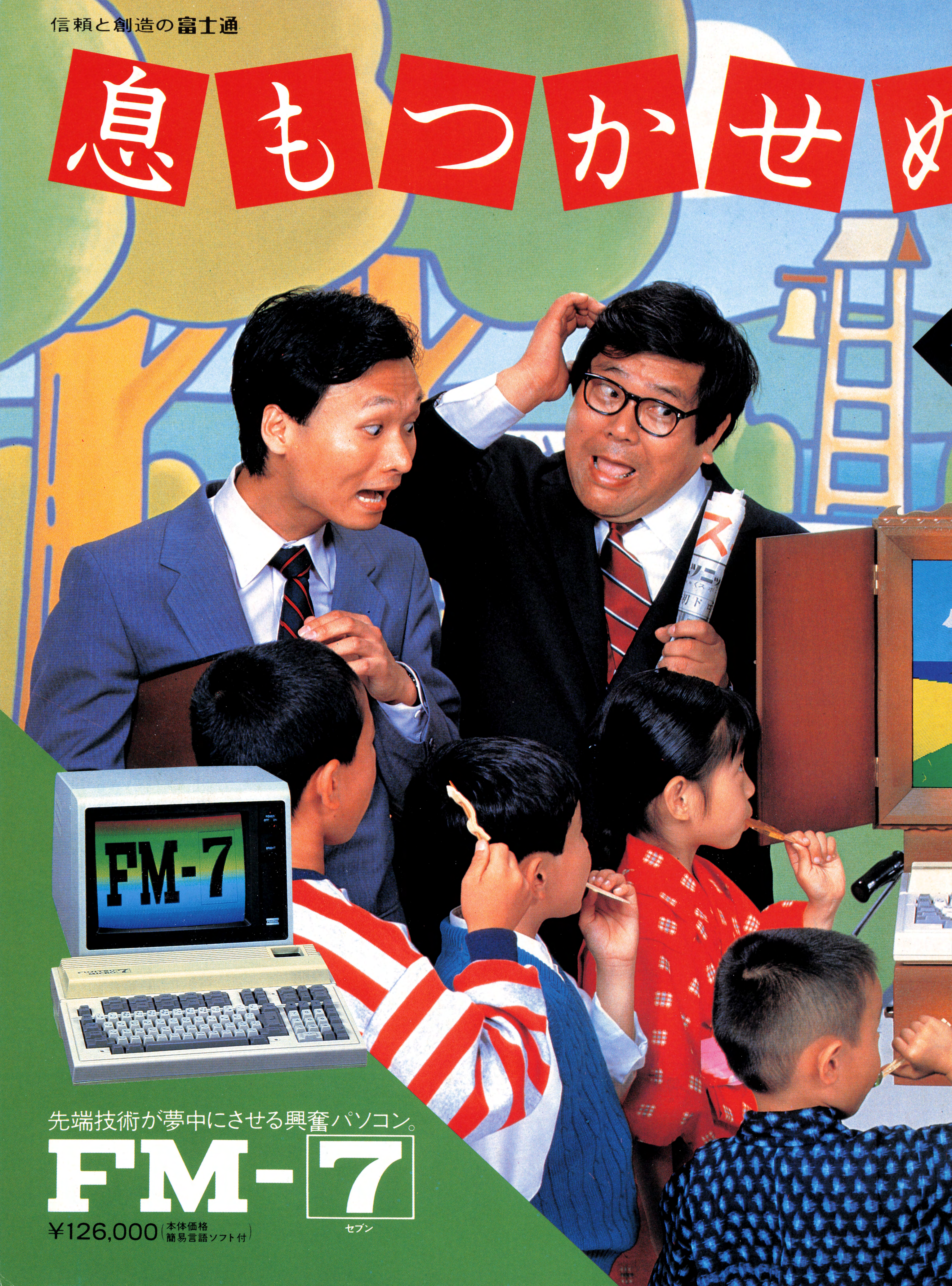
Oh!FM

隔月刊　MICRO 7 8 11　'83年8・9月号

マシン語特集partⅡ

- 6809セルフアセンブラ／実例集
- 6809インストラクションポケットブック

Vol. 4　480円


信頼と創造の富士通
息もつかせぬ
FM-7
先端技術が夢中にさせる興奮パソコン。
FM-7
セブン
¥126,000（本体価格
簡易言語ソフト付）

盛りあがり。
富士通
富士通株式会社

興奮のハード。感激のソフト。

青少年も、ビジネスマンも、ハードとソフトで、盛りあがる。

富士通の興奮パソコン〈FM-7〉。発売以来、もうクライマックスの連続です。豊富な機能のハードに興奮、ますます充実するソフトに感激。ハードがいいから、ソフトもいい。価格が手頃、と話題集中。全国の青少年を、ビジネスマンを、大いに盛りあがらせています。

感激のソフトが、ますます充実

〈FM-7〉で使えるソフトが、どんどんふえています。いろいろなゲームや、ホビー、ビジネス、教育、話題集中のワープロなど、市販のソフトはもとより、〈FM-8〉用の流通ソフトもその多くが、そのまま使えます。

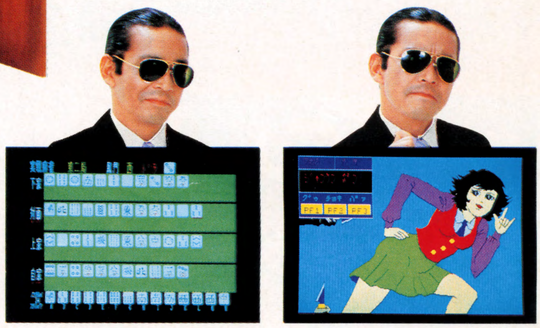
簡易言語がついてくる

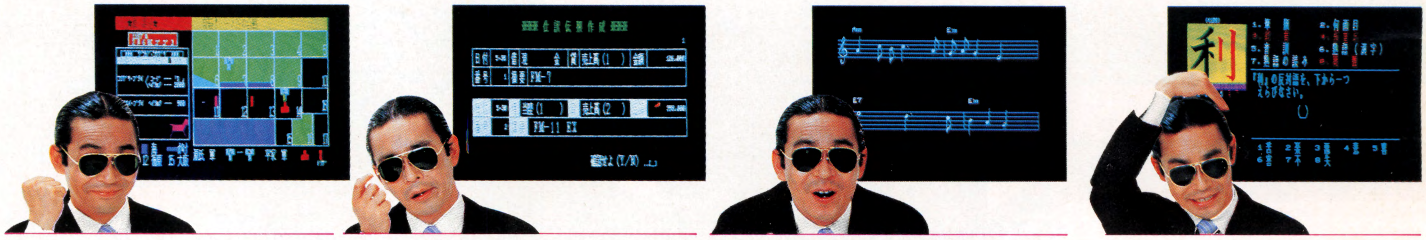
うれしいことに〈FM-7〉には簡易言語が標準装備されています。家庭では住所録や家計簿、またオフィスでは各種資料の作成など実に幅ひろく利用できます。作表や計算、検索や分類、ファイリングなどが自由自在。

●サウンド機能で、ゲーム効果音や8オクターブ、三重和音までの音楽演奏もバッチリ。外部スピーカも接続可能です。

●カラーグラフィック画面は640×200ドットの高分解能。ドット毎に8色まで色指定でき、パレット機能で色交換も簡単です。

●漢字ROMカード(オプション)を装着すれば、漢字、英字、かな、カナ、記号など3,418字種を表示、印字でき、日本語ワープロとしても使えます。





先端技術が夢中にさせる興奮パソコン。

FM-7

セブン

¥126,000
(本体価格
簡易言語ソフト付)

高級ホビーからビジネスまでの多才パソコン。

FM-8

エイト

¥218,000(本体価格)

富士通のパソコンFMシリーズ。

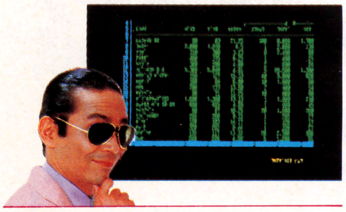
ディスク搭載、16ビットで、ビジネス盛りあがる。

FMシリーズの高級機種<FM-11>は、EX、AD、STの3種類のバリエーションからなり、オフコンにせまる本格派として、EDP部門から現場まで幅広いビジネスユースに対応。パワフルな機能と使い勝手の良さで、先端派ビジネスマンに、エキサイティングな話題をなげかけています。

簡易言語ソフト FM CALCを標準装備

●FM CALCは、わずか18種類のコマンドで操作でき、うち10種類はファンクションキーに設定されているので、ワンタッチで操作可能。

●作表プログラムとして、会計業務、帳簿作成、統計処理など目的に合わせた処理が可能。本格的なビジネス分野に活用できます。



使い易さで群をぬいています

●本体、キーボード分離のセパレートタイプ。ロープロファイル、シリンドリカル・ステップ・スカルプチャー設計のキーボードなど、操作し易さは抜群です。

●CP/M-86の採用によりCOBOL等の高水準言語でも日本語処理が行えます。漢字の拡大、外字登録可能。(漢字ROMカードはオプション)

●メインメモリ128KBを標準実装し、最大1MBまで拡張できます。これによりCP/M-86、MS-DOS、OS-9などのもとで、大容量メモリの必要な高水準言語も余裕をもって利用できます。

●最大640×400ドットのグラフィック表示は、キャラクタとの混在表示も可能。カラー表示は16色。最大12画面のマルチページ機能。パレット機能、スクロール機能、などCRT表示がより多彩になりました。

ニーズにあわせてお求めやすい3タイプ

	EX	AD	ST
メインCPU	MBL8088-2 MBL68B09E	MBL68B09E	MBL68B09E
ミニフロッピィ	1ドライブ	1ドライブ	オプション
システムソフト	CP/M-86 F-BASIC	F-BASIC	F-BASIC (ROM版)
簡易言語ソフト	FM CALC	FM CALC	———

(写真はFM-11 EX フロッピィディスク1ドライブ増設)

ビジネス用途を大きく拡げる高級パソコン。

FM-11
イレブン

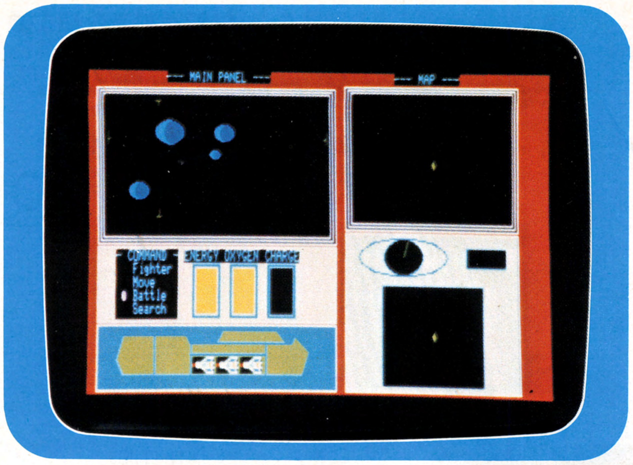
EX ¥398,000
(本体価格・簡易言語ソフト付)

AD ¥338,000
(本体価格・簡易言語ソフト付)

ST ¥268,000
(本体価格)

富士通株式会社 半導体統轄営業部 〒105 東京都港区虎ノ門2-3-13 ☎(03)502-0161
●札幌営業所(011)271-4311 ●東北営業所(0222)64-2131 ●長野営業所(0262)26-8222 ●静岡営業所(0542)54-9131
●名古屋営業所(052)201-8611 ●大阪営業所(06)344-1101 ●広島営業所(082)221-2288 ●九州営業所(092)411-6311
マイコンスカイラブ ──FMシリーズのハードからソフトまで一挙に展示実演。あなたのパソコンのコンサルタントとしてご活用ください
●東京・虎ノ門(03)591-1091/591-2561 ●東京・秋葉原(03)251-1448/251-1449 ●大阪(06)344-7628/341-0486
●名古屋(052)221-6016 ●札幌(011)241-4185 ●広島(082)247-3949 ●仙台(0222)66-8711

富士通



AMPLE SOFTWARE

アンプルソフトウェアはコンピュータと人間との接点を模索します。私達は、ヒューマンインターフェイスによるホームコンピュータ時代をめざしています。

白鳥座GII FM-7（7月20日発売予定）

従来のウォーシミュレーションで複雑だった座標入力と、敵の攻撃中には何も出きないという煩わしさを取り除いた。新タイプのリアルタイム・シミュレーションゲームです。

PC-6001 mk II

パイラス

PC-6001、PC-6001mK II共用
ドットコントロールによるフルグラフィックスアーケードゲームをお楽しみ下さい。 ¥3,000

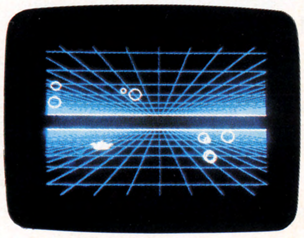
フルハウスIII

PC-8801専用
複雑なルールを完全プログラム化しました。3人で競った結果は、グラフに表示。 ¥3,500

スペースシャトル

FM-7専用
リアルさと美しさ、広大な宇宙をバックのスリルと興奮はFM-7の能力を十分に引き出します。¥3,500

3Dゾーン

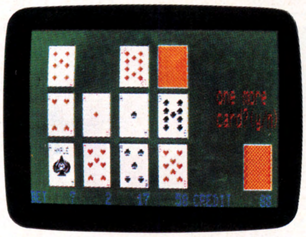


8月20日発売予定
敵を撃破し、低速高速自由自在そして無限の空間へワープ!!
あなたの部屋がコックピットに。

たのしいアンプルパーソナルシリーズ

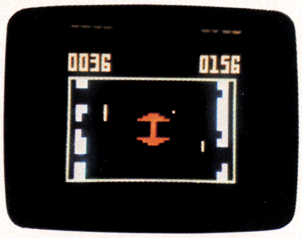
フルハウスI（ポーカー）

PC-8801 3,500円



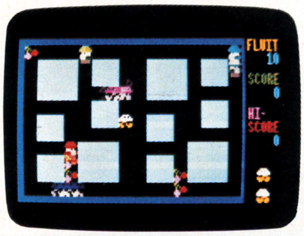
フルハウスII（カブ）

PC-8801 3,500円



リノン

PC-6001 3,000円

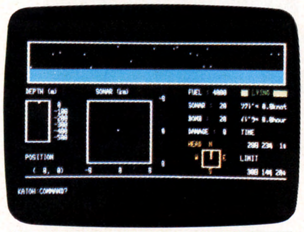


ランダムウォーカー

PC-8001,mKII,8801 3,500円

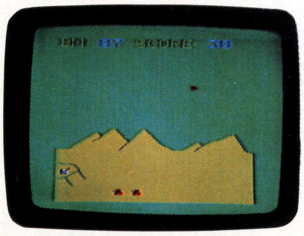
ポッピングパニック

PC-8001,mKII,8801 3,500円



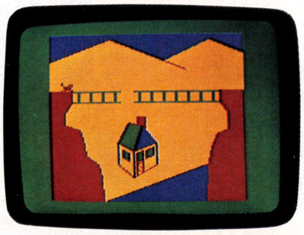
マリアナ海戦

PC-8001,mKII 8801
FM-7,8, CASIO 3,500円



アンプルI

PC-6001(32K) 2,800円



猫の冒険

PC-6001(32K) 3,000円

ゴルダウス

FM-7

AMPLE

マークIIスーパーコンパイラ（PC-8001 MKII専用）高解像度、超高速でしかもN-BASICにフルコンパチブルです。あなたのBASICプログラムがマシン語に自動変換されます。6月発売予定 6,800円。おしゃべりシリーズどんどん開発中!! パソコンが全てを語ってくれる。7月末発売予定 予価 3,000円。
●日本ソフトウェア大賞参加者募集中！ ●技術・開発スタッフ募集中。

アンプルの製品
～アンプルパーソナルウェアシリーズ～
好評発売中
フルハウスI（ポーカー） FM 7
フルハウスII（カブ） FM 7
フルハウスIII（ドボン） FM 7
スーパー暗記術シリーズ MZ700
チェイサー FM 7,8
ゴールド PC8801,8001,MKII
ファイヤーマン PC8801,8001,MKII
各巻 ￥3,500

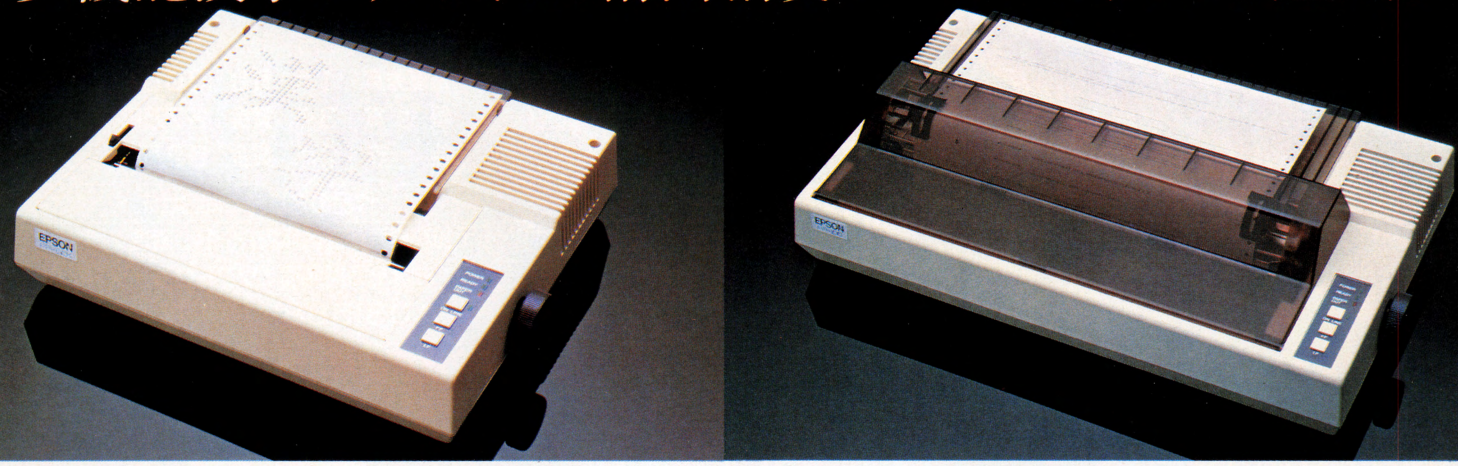
アンプルソフトウェアからのお知らせ！ 当社のソフトウェアは、全国のパソコンショップでお買い求めいただけます。または当社の方へ直接現金書留でお申し込み下さい。

ホームコンピュータエイジ
アンプルソフトウェア株式会社
〒151 東京都渋谷区元代々木町14-3 TEL 03-466-3170(代)

EPSON

プリンタはエプソン

速さと美しさを極めたエプソン卓越の2機種。多機能漢字プリンタ、136桁高精度スーパープリンタ新登場。

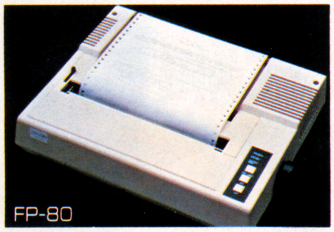


FP-80K 多彩な機能を搭載した高品質漢字プリンタ

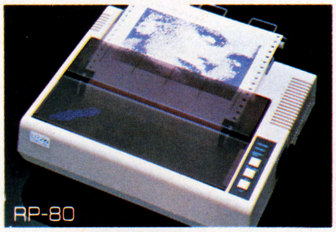
●漢字JIS第1水準約3,500字内蔵の高性能プリンタ●ファンフォールド紙、ロール紙、レター用紙が使える可変ピンフィード、フリクション方式●印字速度、テキストモード160CPS、漢字モード24CPS●通常文字9×9ドット、漢字14×18ドット●豊富なビットイメージモード(9種類)●国際文字(9ヶ国)、ダウンロード文字、さらにプロポーショナル印字など多彩な文字種を生かす豊富なファンクション。¥189,800／¥192,800(PC-8001専用機)／¥193,800(PC-8001mkII／8801専用機)

FP-100 トータルバランスに秀れた万能型スーパープリンタ

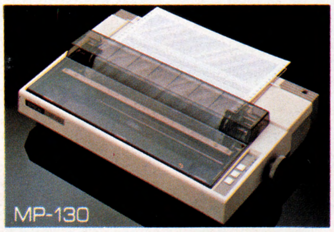
●豊富なビットイメージの高精度図形処理機能。●印字速度160CPS、紙送り速度150ms／行の高速型●インパクト・ドット・マトリックス●超高精度ビットイメージモード(9種類)●通常文字、エリート文字に加え、国際文字、ダウンロード文字の他、プロポーショナル印字などの多彩な文字種(同一行混在可)と豊富なファンクション。●ファンフォールド紙、レター用紙が使えるフリクション&トラクタフィード方式。¥189,800／¥192,800(PC- 8001専用機) ／¥193,800(PC- 8001mkII／8801／9801専用機)



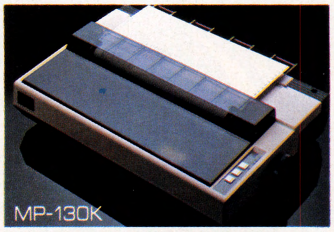
高速印字と多彩な文字バリエーションの強力インテリジェントターミナル●ファンフォールド紙、ロール紙、レター用紙が使える可変ピンフィールド、フリクション&トラクタフィード(オプション)方式●標準インタフェイス：セントロニクス準拠8ビットパラレル●インパクト・ドット・マトリックス●超高精度ビットイメージプリンティング9種類●印字速度、160CPS(テキスト時)●アンダーライン機能、スーパー/サブスクリプト文字、強調文字など多彩な文字種に加え、国際文字印字(9ヶ国対応)、ダウンロード文字(ユーザー定義可能文字)、さらにプロポーショナル印字も可能。また、すべての文字種は同一行混在可。左右のマージン設定、水平タブ、垂直タブなどの機能によりフォーマティングが自由自在。¥149,000／¥152,800(PC-8001専用)／¥153,800(PC-8801／9801専用)



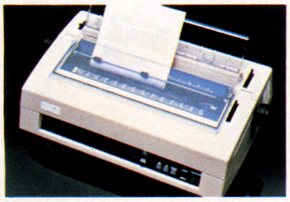
MPの技術が生んだ高性能の経済派プリンタ●4″～10″のファンフォールド紙が使えるトラクタフィード方式●標準インタフェイス：セントロニクス準拠8ビットパラレル●インパクト・ドット・マトリックス●単密度480ドット/行、倍密度、倍速倍密640ドット/行、CRTグラフィックス640、720ドット/行、4倍密度1920ドット/行の高精度ビットイメージプリンティング6種類●印字速度：100CPS(テキスト時)●アンダーライン機能、スーパー/サブスクリプト文字、強調文字など多彩な文字種と豊富なファンクション ¥89,000



高速136桁ビットイメージプリンタの本格派●135字／秒の高速プリンティング●底部からの用紙(ファンフォールド紙)挿入もできる2ウェイ(背面・底面)紙送り方式採用の本格ビジネス仕様●1桁816ドットのビットイメージプリンティング ¥228,000



本格的ビジネスユースに最適の高性能24×24ドット、136桁漢字プリンタ●ファンフォールド紙、ワンシートでB4横サイズまで使える可変スプロケット&フリクションフィード方式●漢字90桁/行、アルファベット・カナ・記号136桁/行●JIS第1水準準拠約4,000字、JIS第2水準準拠約7,000字の漢字ROMを内蔵、JISコードでハンドリングが容易●単密度1224ドット/行、倍密度2448ドット/行の高密度ビットイメージプリンティング ¥510,000(JIS第1水準漢字ROM内蔵) ¥550,000(JIS第2水準漢字ROM内蔵)



DP-20デイジーホイールプリンタ

鮮明な印字と簡単な操作のインパクトプリンタ
●デイジーホイール採用により、ワードプロセシング等に最適●ホイールは600万文字と高寿命●カセット式デイジーホイールおよびリボンメカニズムの採用により、活字・リボン交換がワンタッチ●文字/行：132文字(1/10インチ)、158文字(1/12インチ)、198文字(1/15インチ)●用紙幅：16.5インチ(420㎜)●キャリッジ送り：10文字/インチ、15文字/インチ●ペーパーフィード方式：両方向フリクションフィード¥230,000(シリアル仕様)/¥210,000(パラレル仕様)

各種パソコン対応表

		RP-80		FP-80/100		FP-80/100 (PC-8001専用)		FP-80/100 (PC-8001mkII、PC-8801、PC-9801専用)		FP-80K		FP-80K (PC-8001専用)		FP-80K (PC-8001mkII PC-8801専用)		備考
		CODE	HC	CODE	HC	CODE	HC	CODE	HC	CODE	HC	CODE	HC	CODE	HC	
EPSON	QC-10	◎	●	◎	●	○	●	○	●	◎	●	○	●	○	●	
	QC-20	◎	●	◎	●	○	●	○	●	◎	●	○	●	○	●	
	HC-20	◎		○		○		○		—		—		—		#8145により接続
NEC	PC-6001	○	●	—		—		—		—		—		—		
	PC-8001	◎	◉	○	◉	◎	◉	◎	◉	○	◉	◎	◉	◎	◉	#8135によりハードコピー可
	PC-8001MKII	◎	●	○		◎	●	◎	●	○		◎	●	◎	●	
	PC-8801	◎	●	○		◎	●	◎	●	○		◎	●	◎	●	
	PC-9801	◎	●	○		◎	●	◎	●	—		—		—		
三菱	MALTI16	○		○		—		—		○		—		—		
ソード	M20/23	○		○		—		—		—		—		—		
ダンディ	TRS-80	○		○		—		—		—		—		—		
アップル	APPLEII	◎	◉	◎	◉	—		—		◎	◉	—		—		#8132によりハードコピー可
東芝	パソピア	◎	●	○		◎	●	◎	●	○		◎	●	◎	●	
備考				ダウンロード文字により各コンピュータと文字コード表を一致させることができます												

CODE：文字コード表　HC：ハードコピー　◎文字コード表一致　◉オプションによりハードコピー可　○グラフィックキャラクタ不一致　●ハードコピー可
注）各専用機によりケーブルが異なりますので、ご注意下さい。

フロッピーはエプソン

パソコンのディスクユース時代に、先駆のターミナルフロッピーTF-20。エプソンから

新時代のスリムなフロッピーディスクドライブ装置TF-20。両面倍密度322K×2デッキを内蔵して142,000円(標準機)

<TF-20の特長>薄型・軽量・コンパクト：幅が従来品の1/3の新開発薄型ミニフロッピーディスクドライブ2基を搭載、2ドライブのミニフロッピーディスクユニットとしては驚異的な小型化を実現しました。**高い操作性**：ディスクを挿入しプッシュボタンを押してセット完了。ディスクを取り出すには、押し込まれているプッシュボタンを更に押すと約1cmディスクが取り出し口よりポップアップ。簡単でしかも確実な操作を可能にしています。**ハイコストパフォーマンス**：便利なのはわかっているけれど価格が高すぎて…。TF-20は自社開発ディスクドライブを採用し、高い信頼性、性能を確保しながらトータルコストダウンに成功しました。

<仕様>インタフェイス：シュガートSA400コンパチブル●インタフェイスコネクタ：FRC2-34(DDK)●記録方式：FM、MFM●トラック間セクタ時間：15ms●トラック密度：48TDI●トラック数：80トラック／両面1ドライブ●最大記憶容量：1Mバイト(アンフォーマット)●電源AC100V(50／60Hz)●消費電力：40W以下 ●外形寸法：120W×350D×165H(㎜) ●重量：7.0kg ●価格：エプソン HC-20専用機¥177,000、NEC PC-8001・8001mkII・8801・9801専用機¥166,000、富士通FM-8・7専用機¥163,000
●FM−7用FDD I/Fセット#9040新発売 ¥12,000

TF-20

TF-20ダッシュキャンペーン実施中

●**7月31日(日)まで**エプソン・ターミナルフロッピーTF-20(標準機、PC用、FM用、HC用)をお買上げの方にエプソンオリジナルのフロッピーディスク5枚セットをさし上げています。

エプソン株式会社
●本社／〒399-07 長野県塩尻市広丘原新田80 ☎02635-2-2552㈹

エプソン販売株式会社
本社／〒160東京都新宿区西新宿2-4-1新宿NSビル内私書箱6109号☎03-348-7121(大代)・大阪支店／〒530大阪市北区西天満4-14-3住友生命御堂筋ビル11階☎06-365-5071・名古屋営業所／〒460名古屋市中区金山1-12-14☎052-331-5486・札幌営業所／〒060札幌市中央区北1条西2丁目札幌時計台ビル6階☎011-222-2821・仙台営業所／〒980仙台市本町1-12-12山万ビル6階☎0222-63-3691・長野営業所／〒390松本市深志2-2-9加納ビル2階☎0263-36-7251・広島営業所／〒730広島市中区幟町3-1第3山県ビル4階☎082-223-2811・福岡営業所／〒812福岡市博多区博多駅前3-4-8 ☎092-471-0761
●詳しい資料のご請求は、はがきに住所、氏名、年令、職業、製品名をお書きの上、もよりのエプソン販売までお申込みください。

資料請求券 Oh!FM

ぱそる
ソフトコンテスト
賞金総額
400万円

スーパークレイジーたちの祭典。

締切は秒読みに入った。

応募作品

①ゲームソフトおよび教育用ソフト
（教育用ソフトはゲーム要素の強いもの）
②ゲームおよび教育用ソフトのアイディア
3画面以上のカラーイラストとシナリオを添付すること。両部門とも、自作の未発表作品に限ります。

応募締切 昭和58年9月10日(当日消印有効)

賞

総額400万円

●ソフト部門

最優秀賞1点　賞金　150万円
優秀賞2点　賞金　各50万円
佳　作5点　賞金　各15万円

●アイディア部門

最優秀賞　1点　賞金　30万円
優秀賞　3点　賞金　各7万円
佳　作12点　賞金　各2万円

作品の送り先と問合せ先

〒107 東京都港区赤坂1-5-11　新虎ノ門ビル
㈱ぱそる開発部「ソフトコンテスト」係
☎03(588)-1717　担当／辻本

見やすく、触れやすく、試しやすく、各メーカーの主要機種、周辺機器を豊富に展示。

機能拡大、愉しみもまた拡大

（PC専用）
フロッピーディスク(PC-80S31, NEC)　163,000円
シンセサイザー(CMU-800, アムデック)　65,000円
シンセサイザー(PCS-8007, パックエレクトロ)　24,800円
音声発生(Mr.ボイス, テックメイト)　49,000円
音声発生(読み上げさん, 和幸舎)　35,000円

（PC, FM兼用）
プリンター(GP-250F, 精工舎)　59,800円
プリンター(GP-700M〔カラー〕, 精工舎)　新製品
プリンター(GP-550E〔漢字〕, 精工舎)　新製品
プリンター(RP-80, EPSON)　89,000円
プリンター(FP-80, EPSON)　149,800円
プロッター(マイプロットJr, 渡辺測器)　99,800円
フロッピーディスク(TF-20, EPSON)　142,000円
カラーCRT(FTC-1416, 東映通商)　63,000円
カラーCRT(FTC-1425〔高解〕, 東映通商)　105,000円
デジタイザー(K-510, 関東電子)　148,000円

（FM専用）
フロッピーディスク(薄型フロッピー, 富士通)　167,000円
プリンター(HB27404, 富士通)　89,000円

●PC-8001 mkII

PC-8001mkII(本体)　123,000円

●TF-20

●FM-7

FM-7(本体)　126,000円

●GP-700M

●FTC-1416

●MZ-731

（MZ専用）
カラーCRT(MZ-1D05, シャープ)　69,800円
ジョイスティック(Joy-700, 九十九電気)　5,800円
MZ-731(本体＋カセット・レコーダー＋カラー・プロッタ)128,000円

ぱそるで買うと8つハッピー。

通信販売でも、店頭販売でもぱそるはいま、8大特典で熱気急上昇。

特典1 沖縄の旅10名様ご招待
期間中1万円以上お買上げのお客様を抽選で沖縄の旅にご招待いたします。(期間/58年5月20日～9月30日)

特典2 お買上げ金額ごとに豪華プレゼント
30万円以上　ポケコンPC-1245
20万円以上　ゲーム機器
10万円以上　テープケース
10万円未満5万円以上　生テープ　5本
(期間/58年5月20日～9月30日)

特典3 現金・分割とも**超特価**奉仕

特典4 1万円以上お買上げのお客さまを、ソフトコンテスト受賞イベントに抽選でご招待いたします。
(期間/58年5月20日～9月30日)

特典5 配送費無料

特典6 修理費(配送費とも)無料

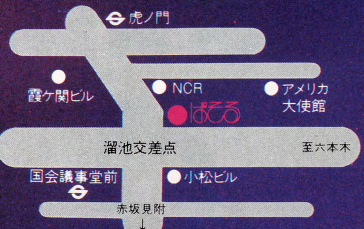
特典7 日本信販との提携ローンで最長36回まで分割OK！

特典8 テレフォンサービス実施
技術指導、ご質問などお気軽にどうぞ。　TEL 03(588)1717
(平日am10:00～pm7:00
土曜am10:00～pm5:00
＊日・祭日はお休みです。)

全国の皆さまへ　ぱそる通信販売

電話またはハガキにてお申し込みください。日本全国どこへでも即納(配送無料)します。ハガキでお申し込みの場合、①機種名、②お支払い方法(現金またはクレジット回数)③氏名・住所・電話番号・年令・職業をご明記ください。

ゆったり、たっぷり
260m²の体験空間。



株式会社ぱそる

本社・ショップ　〒107 東京都港区赤坂1－5－11
(新虎ノ門ビル1F) TEL03(588)1717㈹
和歌山営業所　〒624 和歌山県海南市藤白758番地
富士興産㈱内 TEL07348(2)7332(代)

カタログご希望の方は、ハガキに氏名・住所・年令・職業をご明記のうえ、カタログ請求券を貼ってお申し込みください。

カタログ請求券
Oh!FM 8月号

YDKシステムセンター　新都心・新宿NSビルに430m²のビッグなマイコンショップ

YDKフレッシュアップセール

パソコンデスクがついてくる！

8月31日まで実施中

全館FRESH UP！

全商品ＹＤＫスーパープライスで大奉仕中

ＹＤＫシステムセンターが全面改装。430m²の広いスペースがより一層見やすく、解りやすく、そして買いやすくなりました。おさそい合せの上、お出かけ下さい。

★生活提案コーナー★ＹＤＫおすすめシステム展示コーナー★ＹＤＫソフトバンク★ホビーコーナー★本格ビジネスコーナー★ワードプロセッサコーナー★ご相談カウンター★ビジネスソフト実演コーナー★中古パソコン販売コーナー

YDKオリジナル
パソコンデスク
プレゼント

パソコンをシステムでお買上げのお客様にＹＤＫオリジナルパソコンデスク1台プレゼント。

ご来店サービス
ダーツゲームで
レジャー用品
プレゼント

お買上げ3,000円以上でもれなくダーツゲームでレジャー用品プレゼント。YDK賞はNEC PC-2001！

日本初のフルサポートシステムを採用

- パソコン110番 何でも相談室（会員専用）AM10:00〜PM11:00
- ハードスピード修理、スペア機も用意
- ＹＤＫセミナー・スクール

YDK テレショップ

電話一本・ハガキ一枚で即納品

全国どこでも特別価格でハード、ソフトをお届けします。どこよりも格安です。長期60回まで分割可能なＹＤＫクレジットや会社・商店にとって有利なリースもご利用いただけます。

NECシリーズ

名機PC-8001を磨きあげて画期的大改良

PC-8001MKⅡ

●本体PC-8001MKⅡ　●漢字ＲＯＭボード　●12″カラーモニタPC-8058　●データレコーダPC-6082　●24ピン熱転写プリンタPC-8824（漢字ＲＯＭボード付）

合計標価 ￥500,600

頭金０円　初回6,975円　6,800円×59回　㋭2万×10回

パソコンデスク＋マイコンラボ（36,000円の品）サービス

ＮＥＣのキャリアを生かした高性能。パワフルな16ビットパソコン。

PC-9800シリーズ

●本体PC-9801　●14″高解カラーモニタPC-8853K　●漢字ROMボード　●漢字プリンタNM-9100RO　●ケーブル　●8″FDD PC-9881

合計標価 ￥1,121,500

頭金０円　初回16,835円　11,100円×59回　㋭7万×10回

パソコンデスク＋ソフト５本サービス

184Ｋバイトのメモリを標準実装。すべての機能がハイレベル ビジネスソフトも豊富です。

PC-8800シリーズ

●本体PC-8801　●漢字ＲＯＭボード　●TOEI 14″高解カラーモニタFTC-1435　●プリンタPC-8824　●漢字ＲＯＭボード　●5″FDD PC-80S31　●システムディスク　●カラーケーブル

合計標価 ￥784,800

頭金０円　初回14,810円　9,000円×59回　㋭4万×10回

パソコンデスク＋マイコンラボ（36,000円の品）サービス

シャープ パソコンテレビ X1

コンピュータと映像のクロスオーバー！初のスーパーインポーズ機能 ゲームソフトも日毎に多彩

●本体 ●14″カラーディスプレイテレビ ●グラフィックRAM

合計標価 ￥300,000

頭金０円　初回5,550円　5,100円×29回　㋭3万×5回

パソコンデスク＋ソフト５本サービス

富士通 FM-7

先進技術が無中にさせるヒット作

●本体●シャープ14″カラーモニタ14M-112C●ケーブル　●サンヨーデータレコーダ

合計標価 ￥259,000

頭金０円 初回6,528円　4,500円×29回 ㋭3万×5回

パソコンデスク＋ソフト５本サービス

ミニフロッピーディスク LFD-550PC

＊PC-9801用・PC-8801用・PC-8001MKⅡ用＊
●LFD550PC ●ケーブル ●システムディスケット

合計標価 ￥160,000

頭金０円　初回6,600円　3,000円×59回

ディスケット５枚サービス

新発売コーナー

キャノン X-07

バックにもすっぽり収まるＡ５サイズ。どこでも持ち運び自由の高性能ハンディパーソナルコンピュータ
●本体●サーマルプリンタ

合計標価 ￥89,800

頭金0円 初回5,804円　4,500円×19回

日立 ベーシックマスターL3マーク5

フルグラフィックで高いコストパフォーマンスのフルセット
●本体●14″高解カラーモニタ●3″コンパクトFDD●感熱プリンタ●I/Oカード●システムディスク

合計標価 ￥450,800

頭金0円 初回11,745円 5,900円×59回　㋭2万×10回

パソコンデスク＋コンパクトディスケット5枚サービス

三菱 マルチ8

遊ぶ楽しさに活用する楽しさをプラスした、マルチ8
●本体●14″カラーモニタ●漢字ROMカートリッジ●モニタケーブル●データレコーダユニット

合計標価 ￥281,500

頭金0円 初回8,085円　6,100円×29回㋭3万×5回

パソコンデスクサービス

ナショナル JR-800

32字×8行の大型表示。先進のハンドヘルド

標準価格 ￥128,000

頭金0円 初回5,208円 3,800円×19回 ㋭2万×3回

シャープ MZ-2200

8色カラー対応グラフィック標準装備。自由に組めるコンポタイプＭＺ登場
●本体●12″専用カラーモニタ ●データレコーダ

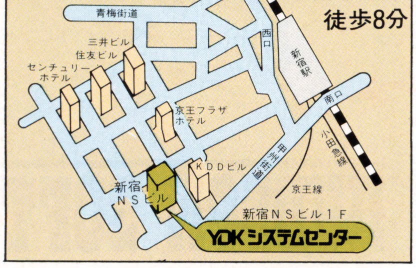
合計標価 ￥249,800

頭金0円初回8,012円5,500円×29回㋭2万×5回

パソコンデスクサービス

●掲載例はほんの一例です。この他各種豊富に取り揃えております。

お出かけ下さい世界の名所　新宿駅南口より徒歩8分



お申し込み方法

全国完全無料配送！（フルサポートのＹＤＫなら後々安心です。）
①商品名（メーカー、型番）②合計標準価格（※標準価格を、記入してもＹＤＫスーパープライスのお取り扱いにします。）③お支払方法（クレジット／現金）④クレジットの時…月払い及びボーナス払いご希望額⑤お名前⑥年齢⑦ご住所⑧電話番号⑨ご職業⑩他、ご意見、ご希望、ご提案などを官製葉書にご記入の上、右記宛ご郵送下さい。

便利なクレジットをご利用下さい。

★クレジットは断然有利、低利のＹＤＫクレジットを！　お金のやりとり一切なしで商品がお手もとに。★記載のクレジット支払い例以外に１回から60回まで可能です。ご遠慮なくお問い合わせ下さい。御振込銀行／住友銀行新宿新都心支店 普通103379 株式会社ワイ・デー・ケー

お問い合せ・お申し込みは

YDKシステムセンター

OF係まで

☎03-342-2441(代)

〒160 東京都新宿区新宿ＮＳビル1階

YELLOW BEE キャンペーン CAMPAIGN

日頃の御愛顧にお応えして……

ダブルチャンス

1回目は25,000名様に当ります。

(A賞．B賞．C賞．D賞．E賞の内抽選で1点が当ります。

2回目は抽選で30名様に特賞が当ります。

A賞
4000個
ハドソン・オリジナル
フローティングキーホルダー

ハドソン・オリジナル
カセット型メモ
B賞
5000個

C賞
ハドソン・オリジナル
三次元ハチスケバッチ
3500個

D賞
2500個

ハドソン・オリジナル
イエローバック(たて51cm×よこ41cm)

E賞 10000個
ハドソン・オリジナル
蜂助バッチ,
すきですサッポロバッチ
2個1組

特賞

■パソコン・ウオッチ 30個

応募者全員からもう一度
抽選で当ります。

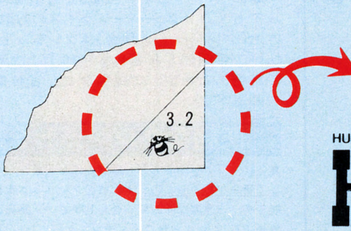
●個々の会計(収支)表示●時、分、秒表示●12あるいは24時間表示選択可能●デイリーアラーム・8桁の計算器、キーサウンド●リチウム電池●ゲーム遊び

● 応募方法　ハドソンの全製品に付いているインデックスを10点以上1口で封筒に入れて送って下さい。(1つの封筒で何口でも申し込めます。)

● あて先　〒102東京都千代田区麹町4丁目7番5号　麹町ロイヤルビル2F
ハドソン東京　YBキャンペーン係

● 締切り　昭和58年7月20日(水)～8月25日(木)の消印有効

● 発表　1回目は賞品の発送をもって発表にかえさせて頂きます。
2回目は月刊ソフトメディア10月号にて発表いたします。



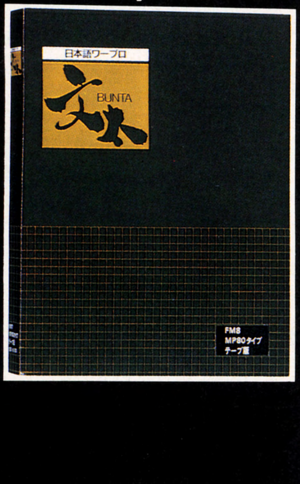
HUDSON GROUP
HUDSON SOFT®

Computer Service

すごい、えらい、するどい。文太 BUNTA!

FM7・8専用日本語ワードプロセッサソフト〃文太〃
アイデア先行の難解ソフトでなく、とても使いやすいのがウリ。
専用機なみの12大機能が「すごい」。
それにもまして、この機能で大納得の9,800円は「えらい!!」
これだけ多くワープロソフトがあるなかで、
いきなりこれが決定版だと納得させる技術が「するどい」。

FM-7 & 8 用
日本語ワードプロセッサ
・¥9,800・

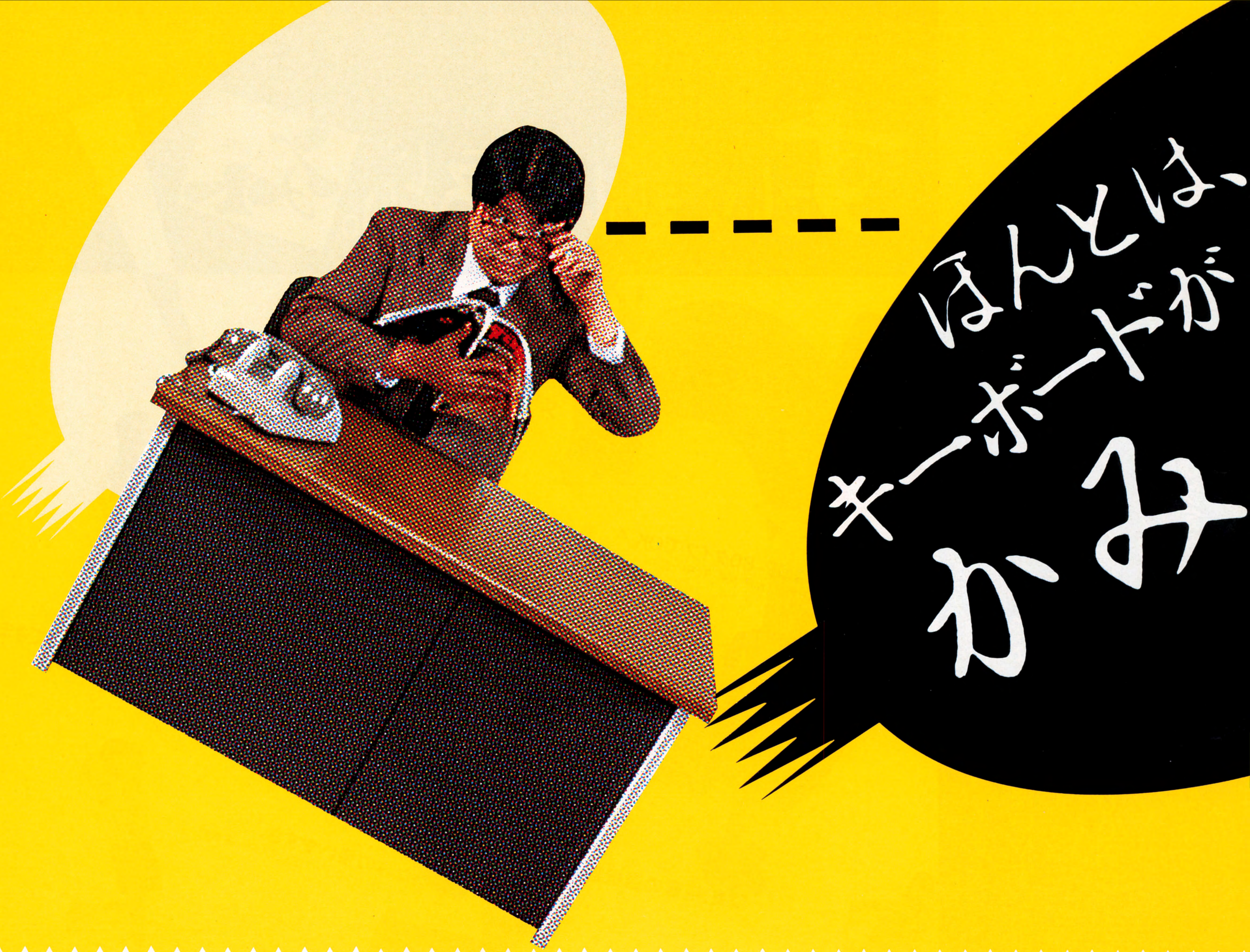


〈お知らせ〉 CSKがIBMと特約店契約を結びました。
●3270エミュレーターの詳しいパンフレットが出来ました。ご希望の方はCSK西日本事業本部システム販売員までハガキでお申し込み下さい。
●情報満載、西日本通販カタログをご希望の方は(通販カタログ希望)と明記の上ハガキにてお申し込み下さい。

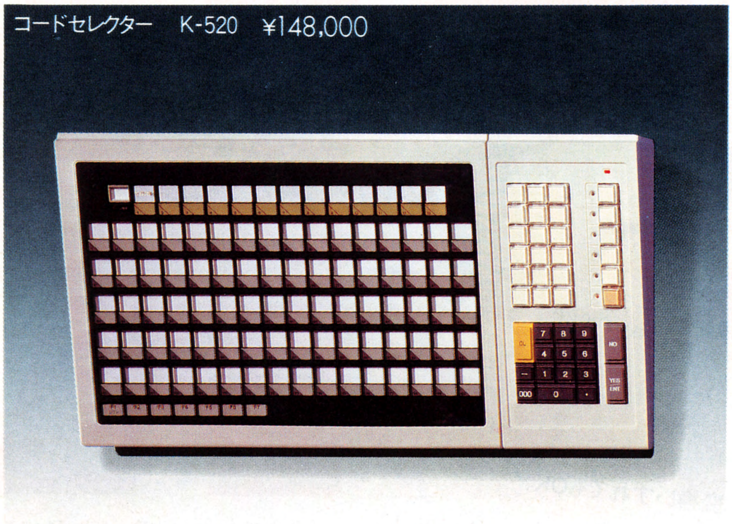
独創・快走!!ソフトウェアーもCSK

マイコンショップCSK
〒530 大阪市北区梅田1-1 3大阪駅前第3ビルB1 ☎(06)345 3351(ショップ直通)

コンピューターサービス株式会社
●西日本事業本部:〒530 大阪市北区梅田1-1 3大阪駅前第3ビル19F ☎(06)345 2851代
●東京・名古屋・九州・日立・広島・ニューヨーク・香港・シンガポール・サウジアラビア



コードセレクターで、パソコンがオフコンを越えられる。



パソコンによるOAは、ますます進んでいます。しかし、だれにでも使えてこそ、パソコンによるOAの真価が発揮されるのですが、現在は日本人特有のキーボードアレルギーや入力の繁雑さのため、多くの問題を残しています。そこで、キーボードにたよらずに、入力できるコードセレクターを開発しました。従来オフコンで使用されているコードセレクターの概念を大きく変え、低価格で高性能、しかも小型軽量であるK-520は、正にパソコンによるOAにぜひなくてはならないものになるでしょう。

Logitec K-520の特長 ●低価格、高性能●小型軽量●シンプルな構造で長寿命●豊富なファンクションキー（42ヶ）●項目数が非常に多くとれる（15,600ヶ）●あらゆるアプリケーションに最適●EIA RS-232Cで簡単にパソコンに接続●ページ切換が高速（約0.5秒）

Logitec K-520のインターフェイス●型式：EIA RS-232C準拠●通信速度：1200,2400,4800,9600Bps（DIP SW切換）●同期方式：調歩同期方式●通信方法：半二重通信方式●情報単位長：8+2（スタートビット、ストップビット：各1ビット）●パリティの有無：DIP SWにより切換 ●コネクター：26pフラットケーブルコネクター/推奨ソケットPS-26SEN-D4P（JAE）又は相当品

Logitec K-520の仕様

スイッチパネル	キー	項目キー 80ヶ（16×5） ページ選択キー 13ヶ ファンクションキー 24ヶ+2ヶ+16ヶ 10キー 14ヶ
	インジケータ	ファンション 6ヶ ページ表示窓 1ヶ
シート/カートリッジ	シート	項目記入カード 13枚 （1040項目/カートリッジ）
	カートリッジ	標準 1本 オプション 14本追加可能、交換が極めて簡単（カートリッジNo検出センサー有り） その他 文字記入：鉛筆色鉛筆 消去：消ゴムです
ページ選択	ページ選択動作 ページ選択動作時間	ページ選択キー、又は、外部コンピュータより、リモート選択可 約0.5sec.
その他	ブザー	シフトおよびシフトロックキー以外のすべてのキー入力時又は、警報時に発音　外部コンピュータから制御可能　音量調整可能
	電源	AC 100V ±10% 50/60Hz30VA
	寸法重量	500W×75H×280D 5.5kg
	寿命	ページ選択動作、シート/カートリッジ、項目キー各100万回以上

業務用、ソフト開発用に、特別価格にて提供させていただきます。
（システム販売部：遠藤・寺内まで）

キーボードから1点1点座標を指定しなくても、図形が簡単に入力できます。

汎用・小型・軽量・安価のデジタイザK-510 ¥148,000

コンピュータにとってグラフィックを扱う事は、ますます重要な仕事になっています。しかし、座標を割り出して、1点1点指定する作業は、時間や手間がかかりすぎます。そこで、デジタイザ。デジタイザは、グラフィックをなぞるだけで、数値データとしてコンピュータに読み込ませることができます。

Logitec K-510の特長

1 安価です。
2 小型軽量です。
3 高精度・高分解能の高い機能を持っています。
4 RS-232C内蔵です。
5 電源内蔵です。
6 自己診断機能が付いていますので、安心です。

●3つの読取りモードについて
K-510は3つの方式で図形等の座標を読取り、入力できます。

ポイントモード
有効エリア上にある図形を、カーソル又はスタイラスのスイッチが押される毎に、その点の座標データが送出されます。

ストリームモード
選択されたサンプリングレートで、データが連続的に送出されます。

スイッチストリームモード
カーソル又はスタイラスのスイッチが押されている間、座標データが選択されたサンプリングレートで、連続的に出されます。

• •

以上、読取中にカーソルが有効読取範囲外にある時は、ブザーが3回鳴り、データは正常に送出されません。

Logitec K-510の仕様

読取方式	絶対座標磁歪方式
有効読取範囲	280×380mm(A3相当)
分解能	0.1mm
ポイント読取り誤差	±0.5mm(A4相当) ±1.0mm(A3相当)/20~25℃
読取材質	紙・マイラー・ガラス等非導電・非磁性体(0.5mm以下)
読取速度	ASCII FORMATの場合 ——2,4,10,20,35,52点/秒 Binary FORMATの場合 ——2,4,10,20,35,70,110点/秒
自己診断機能	デジタイザの電源投入時にROM,RAMチェックを行ない、ROM,RAM不良時は、ブザーを鳴らしてLEDを点滅させます。 ROM不良—READYランプとSTATUSランプが同時に点滅 RAM不良—READYランプとSTATUSランプが交互に点滅
送信データバッファ	1データブロック分の送信データバッファを持つ
受信データバッファ	256Byte
電源	AC100V(±10%) 20VA以下
周波数	50/60Hz
インターフェイス	信号規格—EIA RS-232C準拠 通信速度—150,300,600,1200,2400,4800,9600 BAUD 選択(スイッチにて選択) 同期方式—調歩同期式(ASYNCHRONOUS)
重量	3kg(本体)

関東電子株式会社
POWERFUL & CLEVER
第一販売部〒101 東京都千代田区外神田2-15-2新神田ビル☎03-257-6221(代) システム販売部〒101東京都千代田区外神田2-15-2新神田ビル☎03-257-6291
大阪支店 〒556 大阪市浪速区日本橋3-3-5カトウビル☎06-632-0207(代)
●札幌営業所☎011-832-0131 ●仙台営業所☎0222-33-0257 ●長岡営業所☎0258-32-8888 ●群馬営業所☎0270-23-2301 ●千葉営業所☎0472-48-2955
●多摩営業所☎0423-44-8111 ●町田営業所☎0427-28-8882 ●沼津営業所☎0559-51-2888 ●浜松営業所☎0534-64-2238 ●名古屋営業所☎052-263-1693
●京都営業所☎075-343-0995 ●広島営業所☎082-227-5536 ●福岡営業所☎092-474-5777 ●熊本営業所☎0963-26-1166

FM-11用

高速日本語 ワードプロセッサ JWORD新発売

●CP/M-86・8″(5″も有り)・Ver2.0 ￥82,000

●従来、パソコン用ワープロ・ソフトというと、専用機に比べて操作性が悪いとか、スピードが遅い、辞書が貧弱といったことがあり、ワープロだけは専用機でとお考えの方が多いと思いますが、そういった方は一度、JWORDをご検討下さい●JWORDは、このクラスでは初の本格的な文節変換入力と65,000語という専用機のレベルをも超えた単語辞書を内蔵し、その操作性においても、プログラムを全てアセンブラで記述し、独自の高速メモリ／ファイル管理方式の採用により、格段のスピードアップを実現しています●FM-11では、CP/M-86の元で動作し、ビジネス・プリンタ(MB27402)に対応。

★JWORDは、日本ソフトバンク、マイクロソフトウェア・アソシエイツの各加盟店でも取り扱っております。 ※CP/M-86は、デジタル・リサーチ社の登録商標です。

★文節単位によるカナ漢字変換

〈例〉モウシアゲマス。→変換→申し上げます。といった、文節単位の変換を実現！

★65,000語の単語辞書内蔵

なんと65,000語の単語辞書と7,000語の漢字辞書内蔵。ユーザ単語も10,0000語！

★ローマ字、混在など4入力

入力は、便利なカナ漢字変換で、カタカナ、ローマ字、混在の4入力方式に対応！

★強力な作表、演算機能装備

矢印キーにより画面上に作表が可能。また、電卓なみの簡単さで、四則演算が可能！

★アセンブラ記述で超高速

プログラムは全てアセンブラで記述され、独自のメモリ/ファイル管理で超高速！

●関連会社 E&M イー・アンド・エム(株)
エレクトロニクスと人間の調和

ACEL エイセル株式会社
〒102東京都千代田区麹町3-4-7ビルディング啓 ☎03(263)6421(代)

資料請求券 Oh!FM8係

マイコンショー'83に何をみた!?

取材：
慶応大学メモリーマイコンクラブ
田奈真実

すでに開催から2か月もたってしまったが，今年も「マイクロコンピュータショウ'83」が5月25日～28日，相も変わらず東京流通センター（羽田空港に近い平和島）で行われた。恒例のイベントとなったこのショー，マイコン関係では最大級のショーで，メーカーが新製品を発表することが多く，我々その筋の者(？)にとっては見逃せない。一昨年は，我らが崇め奉るところのFM-8，当時はその安さがショッキングだったコモドールのVIC-1001，タンディのTRS-80カラーコンピュータなどが発表されている。

昨年は，パソコンブームを受け，メインフレームが続々と出現した。シャープのMZ-2000，松下の低価格パソコンJR-200，初のハンドヘルドコンピュータとして発表されたエプソンのHC-20，パソコンとしては初の16ビットだったNECのN5200，日立の16000，カシオのFP-1100／1000／5700などである。

最近では，そうでない例も出てきてはいるが，春のマイコンショー，秋のデータショーに照準を合わせて新機種を開発するメーカーは多い。さて，今回のマイコンショーでは，どんなメインフレームが発表されたのであろうか（と言っても読者諸氏は既にご存知かとも思う)。というわけで，今回で8回を数えるマイコン界の大イベント，マイコンショーを取材してみた。

12万円を切ったうえ，カラープライオリティや6和音など独特の機能を持つパソピア7。このショーで一番注目すべきマシン（だよ，きっと）。

モトローラの 68000 を使ったソードM68。日本では 68000 を使っているメインフレームはまだ少ない。

12万円クラスのパソコンが続々と…

FM-7が12万6,000円で発売されて以来，この価格帯がパソコンの一つのクラスとなった。このクラスの特徴は640×200のカラーグラフィック機能を持っていることで，既にNEC（PC-8001mkII），日立（レベル3 MARK 5），カシオ（FP-1100）が出ている。今回のショーでは，他に3社がこの価格帯で新機種を発表した。

この中での一番の出物は東芝のパソピア7であろう。このネーミングからFM-7を思い浮かべる方もいらっしゃるのではないだろうか。筆者はそう感じたのだが，機能は確かにFM-7に似ていて，というよりもFM-7の機能を完全に包含した上位機種のようだ。CPUはZ80A，RAMはVRAMと別に64Kを持ち，VRAMが56Kバイト，グラフィックは640×200の8色，サウンドが同時6音2ノイズの構成をとっている。つまり，機能の面からだけ見れば，FM-7よりPSGが1つ多いだけだと極論してしまうこともできる。つまり，この12万円クラスではFM-7の機能が最低限必要であるということなのかもしれない。

今までパソコンとしてはMULTI 16しか出していなかった三菱も，その下位機種としてMULTI 8を発表した。CPUはやはりZ80A，640×200のカラーグラフィックを持ち，サウンドは3和音で12万3,000円である。ここまでがいわば標準的な機能。このマシンの特徴は唯一，RS-232Cを内蔵していることだ。

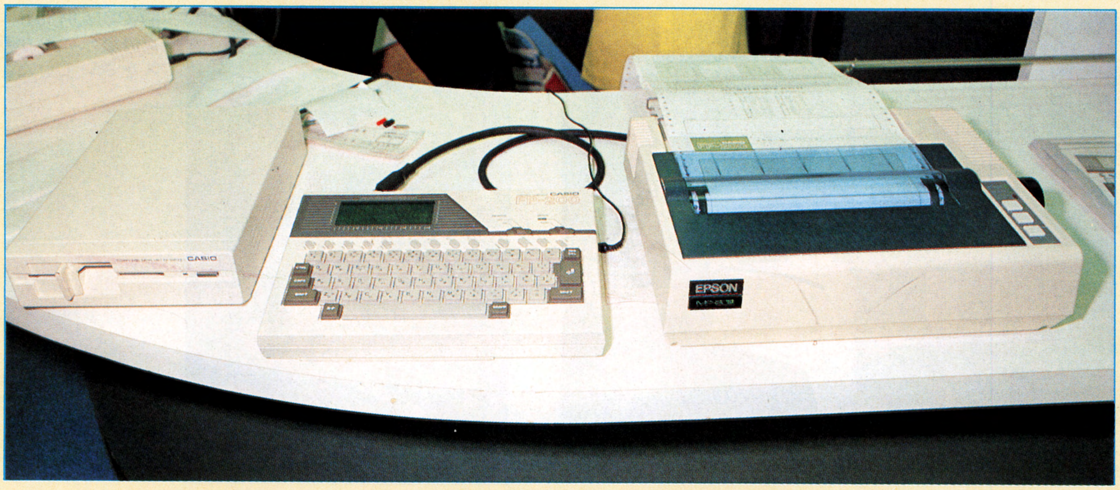
あのシャープからは12万8,000円でMZ-2200が発売された。型番が示すとおり，MZ-2000系統のマシンで，MZ-2000からCRTディスプレイとカセットドライブを取り外したものになっている。グラフィックVRAMを標準実装とした以外，ほとんど変更はなく「いかにもシャープ」らしい新製品の出し方である。当然のことながら，ROMにBASICを持っていない。

この東芝，三菱，シャープに，既に出している富士通，NEC，日立，カシオを加えると，12万円クラスのパソコンが7機種も出たことになる。機能的にはFM-7を標準とし，みな似たり寄ったりといったところで，やや供給過剰というか過当競争のきらいが見えはじめている。その中で注目すべきなのが，今回発表のパソピア，マルチでは既にゲームのデモをしていたことである。これまでは，デモプログラムを走らせるのが通例で，ゲームなどのアプリケーションソフトが走っていることはなかった。日立，シャープは既存の機種と互換性があ

NECの9800にトラックボールを接続したシステム。トラックボールやマウスはパソコンのだいじな入力装置。これを使うマシンはますます多くなるだろうね。

NEC・PC88上の「新日本語入力方式」のキーボード。手の形にキートップが配置されている。

カシオのハンドヘルドFP-200。やはりカシオが出しただけあって69,800円とハンドヘルドでは破格の値段。ただプリンタは本体内に内蔵できない。でも安いから許しちゃう。



るためソフトの蓄積もあったろうが，東芝，三菱は発表前にソフトハウスへマシンを出し，開発してもらわなければならなかったのだから，相当な力の入れようと見ることができる。これは，とりも直さず，12万円クラスの競争の激しさを示している。

そのほかのメインフレームは…

サンヨーがMBC-55という16ビットマシンを出した。CPUは8088（あれっ！ このCPU 8ビットじゃないの？）のクロック3.6MHzでディスク内蔵型。5インチのディスク（なんと，片面倍密‼）が1ドライブ標準実装されている。と，ここまでは普通（？）なのだが，何といってもおもしろいのはCRTへの信号がNTSCのコンポジット信号になっていることである。そのため画面は，ほとんどアップル気分になっている。が，オプションによってRGB出力のCRTインタフェースや5インチ両面倍密倍トラのディスクなども可能である。

ビデオゲームでは，トップクラスのメーカーであるセガが29,800円でキディコンピュータを発表した。このSC-3200は，このクラスのパソコンとしては標準的な構成をとっているが，あのセガが出したということと，3万円弱という低価格の2点が注目される。

さて富士通はどうかな…

富士通は新製品であるオフィスパソコンFACOM-9450IIと，デバイス関係の展示，FMシリーズ関係の展示となっていた。新機種の発表が特になかったため，ソフトや周辺機器の紹介をしていたのだが，それについてふれてみよう。

OS-9

一番目立っていたのが，FM-11，FM-7上のOS-9であった。FM-11をホストにし，その端末としてFM-7を使用したシステムで，画面を複数に分割するマルチウィンドウ機能のデモが行われていた。マルチウィンドウ機能とは，分割した画面を別々のタスクに対し割り当てることで，これによりマルチタスクの実行がわかりやすくなるものである。この大きな特徴をもつDOS，OS-9は，6809用にマイクロウェア社が開発したもので，これについては本誌創刊号を参照していただきたい。FM-8，レベル3用のOS-9は販売されていたが，FM-7，

とにかくユニークなサンヨーの16ビットパソコンMBC-55。標準でテキスト画面が36字×25行，CPUの8088はクロック3.6MHzなど，とにかく型破りなマシンだ。

PC-8201を使ってアンケートを取るNECの人。中には勝手にプログラミングを始めるクラーイ人，いなかったのかなー？

FM-7，11のOS-9のデモ。マルチユーザー，マルチタスクを十分に活かしていた。

FACOM 9450Ⅱ。このマシンの展示にかなりのスペースをさいていた。

11用はこれが初めての発表である。

ローカルネットワーク

FMシリーズ（8，11）用のローカルネットワーク（LAN）のデモが行われていた。このネットワークは，アメリカのコーバス社が作ったOMNINET（オムニネット）とコンパチブルである。またネットワーク上に最大64台の端末を接続することができる。

128Kバブルメモリ

かねてからアナウンスのあった128KBのバブルメモリとそのインタフェースの実物が，やっと我々に姿を見せてくれた。現在FM-8には32KBのバブルメモリがオプションで使えるが，32KBではあまりに容量が少なすぎ，バブル特有の不揮発性，高速性を活かすことができなかった。この128Kによって，バブルメモリの使い勝手がかなり改善されるだろう。

Logo

FM-7，8用のFM-Logoが発表された。今回のショーでは，ソニー，シャープもLogoを発表しており，今後の動向が非常に注目されるところだ。LogoはMIT（マサチュセッツ工科大学）で開発されたプログラミング言語で，LISPにタートルグラフィックスが付いたような言語である。LISPにかなり似ていて，人工知能型の言語となっているため，人工知能（AI）の実験などを行える。また，自然言語に近く，タートルグラフィックスが使えることからCAIなど幼児のコンピュータ教育用にも期待されている言語である。

配布されたパンフレットによれば，供給は今年の9月から開始されるとのことだ。

今年のマイコンショー

あまり目立ったメインフレームの発表がなかった今年は，アプリケーションソフトや言語などのソフト，周辺機器などのハードの発表が多かった。NECが出した電話と合体したパーソナルカプラなど大変興味深いものであるし，Logoに代表される言語・アプリケーションソフトの発表には，実際にマイコンを応用していくという次の時代を感じることができる。INS構想などインテリジェントなツールが現実に必要とされている今，メインフレームの発表が少なかったことが，殊更，次の世代のマイコン界を感じさせるのである。

＊　＊　＊

128KBバブルメモリ。左がバブルホルダ，FM-8に接続している。静かで高速，電源を切っても内容を忘れないという特性からユーザーが増えるかも。

タートルグラフィックスを使ったFM-Logoのデモ。タートルは三角形じゃなくてほんとの亀さんの方が，絶対かわいいんだけれど。

Oh!FM

Vol. 4
'83年8・9月号

目次

マシン語特集 partII

1 目的・予算別マシン語開発ツール　平野 剛　18
2 初心者向けプログラム開発話　本GO誠司　22
3 6809セルフアセンブラ　桑原岳夫　31
4 マシン語プログラム実例集
(1) ソート&画面操作　鶴岡哲明　37
(2) 簡単なサブシステムの駆動法　吉田健司　45
5 8ビットCPU談義　真&文彦+α　55
付録：6809インストラクションポケットブック

ソフトから飛び出せ！ハード入門
はじめて握るはんだごて①
――インタフェースを作ろう
志沢雅彦　61

FM-7専用ページ
マシン語でPSGをコントロールしよう！
PSGサウンドエディタ
小高康弘　89

ぴかぴかのパソコン一年生のページ
画面制御から入るF-BASIC入門
石原宏明　100
おもしろパソコン用語集　98
FMシリーズマニュアルバグ情報　72

ビジネス志向派のページ
FMCALC――その基本機能と応用
岡部 正　74
PCA財務会計使用記
田中高志　114

ゲームリスト　スペースシャトル　81

連載講座
ソフトウェア設計法入門講座(最終回)
武原 宰　117
おわび：「F-BASIC中級入門」は都合により今号は休載いたします。

情報のページ
マイコンショー'83に何を見た?!
田奈真実　13
新製品情報　44
プレゼント情報　113

読者と共に作るページ
ショートプログラム集　109
Q&A　80
READER'S AREA　112

● EDITOR'S ROOM　144
● 広告索引　144

・CP/M, CP/M-86, MP/MはDigital Research社
・MS-DOSはMicrosoft社
・FLEXはTSC社
・UNIXはBell Laboratories
・UCSD p-Systemはカルフォルニア大学理事会
・OS-9はMICROWARE社
・SB-80, SB-86はLIFEBOAT ASSOCIATES
・WORDSTAR, MAILMERGE, SPELLSTAR, WORDMASTER, CALCSTAR, DATASTAR, SUPERSORT, INFOSTARはMicro Pro社
の各メーカーの登録商標です。
その他プログラム名，システム名，CPU名は一般に各開発メーカーの登録商標です。
本文中では"TM"，"Ⓡ"マークは明記していません。

★本誌掲載のプログラムは，個人で利用するほかは著作権法上 無断複製を禁じられています。

マシン語特集 part II

1 目的・予算別 マシン語開発ツール

平　野　　剛

Oh! FMをはじめ，いろいろな雑誌のマシン語入門記事を読むと，僕らも何んとなく6809のマシン語ができるんじゃないかな，なんて思うのだけど，それじゃヨシ，マシン語プログラムを書こうっていうと，これがなかなか書けない。なぜかっていうと，それは道具，ツールがないからなんだ。基板を作るときのはんだごて，犬小屋を作るときのトンカチやカンナみたいに，マシン語プログラムを作るときには，道具――マシン語開発ツール――っていうものがいる。そこでこのコーナーでは，そのマシン語開発ツールの目的別・予算別の紹介をしましょう。

マシン語をやりたくなる動機

お金をためて，FM本体とカラーモニタかグリーンモニタをやっと手に入れた，もう少しお金があればプリンタかディスクが買えるんだけど……っていうのが今の僕たちの本音だよね。はじめの1か月くらいはそれでも，雑誌に載っているBASICのプログラムリストを打ち込んだり（もちろん友達と分担して打ち込んでマージして），関数のグラフを描くプログラムを作ってテープにセーブしたりして，結構満足できているんだけど……。どうも気になりはじめるんだよね，どうしても欲しくなる，それがスピード。そうスピードなんだよね。

ゲームをやっても三次元グラフィックに挑戦しても，やっぱりスピードがネックになってどうもおもしろくなくなる。それで最初は，「マイコンってのは遅いんだよ」なんて専門家のおじさんの言っていることを信じてしまいそうになる。でも，あのインベーダやギャラクシアン，ちっとも遅くない。ゲームセンターのゲームだって，僕らのマイコンより遅いCPUを使ってるし，空対艦ミサイルみたいに超高速処理が必要そうなものにだって，8ビットCPUと計算速度はたいして変わらないCPU（信頼性は高いけど）が組み込まれている。人の目や耳でもって，「これは速いな」って感じるようにするポテンシャルは6809にだって十分ある。だって，クロック8MHz(FM-7)とか4MHz(FM-8)で動いているんだ。1秒間に800万回とか400万回なんだ！　6809は速いはずなのに，それでは，なぜ自分で作ったプログラムは遅いのか。それはBASICで書いているから。

というわけで，何んとしてもマシン語！

そこで，マシン語のプログラムを作ろうと思っている人に，マシン語開発ツールの紹介となったのだけど，ひと口にマシン語開発ツールといっても多種多様，ソフトもハードもあるので，FMシリーズで現在使えるマシン語開発ツールを紹介しよう。

ただし，「この開発ツールがいいですよ」といっても，そんなツールは当然お金がかかる。楽に出せる人もいるし，とてもとてもという人もいる。そこで，予算別に分けてみました。自分で出せる金額と，ここまではやりたいという目的を，じっくり付き合わせてください。

ぜんぜん予算がないあなた

本体とモニタとカセットだけはあるが，あとは何もない，ソフトを買うお金もないというあなた。あなたには，あの有名なハンドアセンブルというのをおすすめします。

これは，紙と鉛筆と6809マシン語命令一覧表（本誌今号の付録，あれです）を用意すればできる。まずプログラムをニモニックなど(たとえば，LAD $2000とかPSHS X，Yとかいうやつです)で書いて，命令一覧表から対応するマシン語コードを探しだしていく。だれでも一度やってみるといいと思うけど，これは本当に大変な作業なんだ。

ツール（道具）というのがなぜあるのかっていうと，プログラム作りの能率を上げるため，というのが正解。上のハンドアセンブルという方法は，実に能率が悪い。たとえば，

① ニモニックからマシン語コードを探したり，命令の後半部分であるオペランド部の計算（ビット計算やブランチの飛び先番地計算）をしたりするのは，時間はかかるし，また誤りもおかしやすい（6809では同じ命令でもアドレス指定の仕方の違いで，マシン語コードがいくつもあるんだ）。

② できあがったマシン語プログラムはBASICのPOKE文などや，MONコマンドを使ってメモリ上に置いていくのだけれど，このときに誤りをしやすい。

③ さて，メモリ上に置いたプログラムがちゃんと動くかは，ただそれを走らせてみるしか知る方法がない。BASIC

のEXEC文か，MONのGコマンドで走らせてみる。プログラムにバグがあってもBASICのように停止して教えてくれたりはしない。それどころかリセット以外では止まってくれず，プログラムもおシャカになる。

など，ひどく非能率的なんだ。一人の人間のパワーを一定量だと考えると（もっとも悪条件だと馬鹿力が出るともいうけれど），こういうことに労力が使われる分，ほかの配慮（プログラム全体の手順とか，実行速度を上げることなど）へまわす分がなくなっていく。それに，もっと大切なのは，プログラムが大きくなればなるほど誤りの発生件数はどんどん上がる（発生確率が一定でさえも）し，プログラムをメモリ上に置くなどの手間も増えていくっていうこと。そして，ある程度以上大きなプログラムは，結局，実際は作れないということになってしまう。

こうして，能率を下げ，挙句にプログラム作りを不可能にしてしまう原因（上の①，②，③など）をよく見ると，ほとんどのものが，本当は計算機が得意なことを人間がやっているといえる。①のコード探しや計算，②のメモリへの転送など，計算機に任せれば誤りなく一瞬でやってくれるのに，なぜ人間がしこしこやらなければいけないか，そう，それはマシン語開発ツールがないからなんだ。

そこで，予算ゼロの人にも何かないかというと，ありました。ミニアセンブラとかミニ逆アセンブラっていうやつで，こういうのは某雑誌のバックナンバをたぐれば，6809用のリストが必ず載っている。本誌にもあるけど，これは桑原君がひどくがんばっちゃって，ミニアセンブラっていうよりほとんどアセンブラになっている。そこで，こういう雑誌などに掲載されているプログラムリストを，半日もかけて打ち込んで，テープにセーブしてしまえば，もう自分のツールになってしまう。

さて，アセンブラだけど，もう知っているよね，そう，ニモニックとかオペコードというアルファベット3～5文字で表される命令と，オペランドという目的とするアドレスやデータを書き並べると（この言語をアセンブリという），マシン語に変換してくれるのが，アセンブラの基本機能なんだ。つまり，アセンブリ言語をマシン語に自動的に直してくれるのがアセンブラというわけだ。この基本機能だけをやってくれるのがミニアセンブラ。FLEXやOS-9で走るようなアセンブラは，マシン語プログラムを作りやすくするために，ほかにいろいろな機能がついている——つまり，より良いツールである，ということ。

ミニアセンブラの使い方は，とっても単純。どのアドレスからプログラムを始めるか教えて，あとはアセンブルを打っていけば，メモリ上に変換したマシン語を作ってくれる。これをSAVEMでカセットにセーブしてしまえば，マシン語プログラムのできあがり。

次はミニ逆アセンブラ。逆アセンブラのやっぱりミニ版。逆（ディス）アセンブラっていうのは，アセンブラの逆，つまり，マシン語を読んでそれをニモニックに変換してくれる。これもやっぱりFLEXやOS-9で走るような本物は，ツールとしてとっても良くできている。ミニ逆アセンブラの機能は，ここから逆アセンブルしなさいっていうアドレスを教えると，どんどんマシン語をニモニックに訳していくだけ。

ミニアセンブラもミニ逆アセンブラも単純な機能しかないけれど，何もないよりずっといい。ミニアセンブラを使えば，ハンドアセンブルよりずっと速くプログラムが作れるし，それに，誤りをする確率がぐーんと下がる。なぜかといえば，前述の①の表からのコード探しがなくなる，②のメモリへの移しかえだってなくなるからね。

ミニ逆アセンブラも，作ったマシン語プログラムがちゃんとできているか，アセンブリ言語に変えてみてチェックするとか，BASICのROMをアセンブリ言語に訳させてマシン語のお手本（あまり良いお手本と言えないものも多いけれど）として，解析したり，真似したり，など使い出がある。

でも，ハンドアセンブルはもちろん，一般にミニアセンブラと言われてるものを使ってのマシン語プログラム作りも，決して能率的なものではないし（①の飛び先番地計算は人間がやる。これがひどく面倒。もっとも本誌に載っているのはこれをやってくれる），作ったプログラムを積み重ねて，一つの大きなプログラムを作りあげるといった機能がない（BASICのLOADMで一つ一つロードしてつなぎ合わせていくしかない）ので，BASICから呼ばれる小さなサブプログラム程度のものを作るのが限界で，それ以上，例えば，オールマシン語のゲームや，システムモニタを作ろうとすると，どっと負担が増すこと請け合いだ。

一万円ぐらい予算があるあなた

やっぱし本体とモニタとカセットしかないけれど，もう少しお金をもっているっていう人，カセットテープに入って売っているコンパイラがあります。

そもコンパイラとは何んぞや。速いマシン語プログラムを作るには，必ずしもアセンブラを使う必要はないのです。たとえばBASICコンパイラといえば，BASICで書いたプログラムを入力してやると，それをマシン語に翻訳して，マシン語のプログラムを出力してくれる，というすごいツールなのだ（囲み記事『**インタプリタとコンパイラ**』参照）。

一番便利なのは，いま使っているF-BASICを6809のマシン語に翻訳してくれるようなコンパイラなんだけれど，残念，F-BASICの機能が多すぎて，ROMモードのままで走るカセット版F-BASICコンパイラは，今後も現れないだろうって言われている。

でも，もっとシンプルな高級言語のコンパイラはすでにいくつかカセットに入って売っている。

最も古典的なのがCOMSOLコンパイラ。PASCALに似た言語で，BASICの注釈文（REM文）の形でCOMSOLのプログラムを書いて（これは普通のBASICプログラムを作る時と同じ），今度はCOMSOLのコンパイラ自身をLOADMで，メモリ上にもってくる。EXECでコンパイラを実行させると，コンパイラは，先ほど書いたCOMSOLのプログラムを読んで，それを6809のマシン語に翻訳して，オブジェクトモジュール（これはマシン語プログラムなんだけれど，このままでは走らない）という形でカセットテープにセーブしていく。コンパイラの仕事はこれでおしまい。COMSOLでは，このあと，このオブジェクトモジュールをLINKERっていうツールに通さないと，

即実行っていうマシン語プログラムにはならなくて，少し面倒くさい。

こういうコンパイラはほかにもKコンパイラっていう有名なのがあって，こちらは文法がよりBASICに近くなっている。COMSOLもKも高級言語と呼ばれているとおり，アセンブラを使って書くよりずっと簡単にプログラムが書ける。

ところで,「高級言語って何？」と言う人のために一言。高級言語っていうのは，計算機のしくみについて気にしないでプログラムを書ける言語なんだ。だからBASICも高級言語だ。いま話しているCOMSOLもKも計算機のしくみ（たとえば，プログラムをどのアドレスから置くかとか，今どのレジスタに何が入っているかとか，もちろん今どのCPUのためのプログラムを書いているか，など）なんて，全然気にしなくていい。だから，ずっと簡単にプログラムが書ける。

COMSOLやKは，この「プログラムが簡単に書ける」以外にも，前述の①，②の問題点は全くないし，③についても，ミニアセンブラと違って，ソースプログラム（コンパイルする前の，COMSOLやKで書いてあるプログラムのこと）が残るので，バグ探しや，その修正をしたあと，もう一回コンパイルすれば，ちゃんと走るマシン語プログラムを作りあげられる。

さて，このテープ版のコンパイラを使って何ができるかというと，時間を惜しまない人なら，何んでもできてしまう。ゲーム作り，システム作り，はたまたセンサをつなげて計測に，ロボットの制御などなど。だけど，ゲームにしてもシステムにしても，大きなプログラムは突貫でできあがってしまうものではなく，基本的なプログラムの積み重ねなんだ。そこでとても大切になってくるのが，ソースプログラムの蓄積やその整理。そうなってくると，カセットしかない場合には，ちょっと大変になってくる。

また，同じ1万円くらいの予算でも，ただマシン語の速いプログラムができればいいっていうのではなく，CPUや計算機の勉強のために，しっかりアセンブリをやりたいっていう人のためには，EXCEL-9っていうマシン語開発ツールがある。

このEXCEL-9っていうのは，なかなか歴史のあるプログラムで，もともとAPPLEの6809カード用のツールだったのを，FMシリーズでも走るようにしたもの。機能としては，先ほど述べたミニアセンブラやミニ逆アセンブラとモニタを一つにしたものっという感じで，マシン語を勉強したい人には，とても向いているかもしれない。EXCEL-9自体もマシン語で書いてあるから，いいお手本にもなる。

桑原君のアセンブラも，ラベルが使えたり，ソースリストが残るなど，BASICなどで書かれたミニアセンブラなんか比べものにならないいろいろな機能がついている。だから，本誌の今号は一万円ぐらいの価値がある（？）んだ。

15万円以上予算があるあなた

これだけ予算があれば，ディスクが買える。ディスクがあれば，ソースプログラムの整理も簡単だし，それに素敵なオペレーティングシステム（OS）を走らせることだってできる。ディスク＋FLEXなら秋葉原で15万で足りるし，OS-9ももうすぐ数万円くらいで手に入るようになる。

テープで困るのは，ランダムアクセスができないことと，転送速度が遅いこと。コンパイラやアセンブラを使うときは，本当はソースプログラムもコンパイラが作り出すマシン語プログラムもメモリ上に置かず

インタプリタとコンパイラ

コンピュータは0，1でできた2進数マシン語しか理解できない。でも，BASICプログラムでも動いているって？　実はFMシリーズをはじめパソコンは，BASICインタプリタというものを通してBASICをマシン語に変換している。ただインタプリタは，BASICプログラムを1行実行するたびにマシン語に訳していくため，スピードが遅い。これに対しコンパイラは，まず全部をマシン語に訳してしまう。だから，いざ実行となれば，ずっとスピードは上がる。

インタプリタは通訳，コンパイラは翻訳，と思ってもらえばいいだろう。

インタプリタは通訳するもの

コンパイラは番羽訳するもの

に，外から読んできて，コンパイラでマシン語にして，また外に出すのがいい。なぜかというと，ただただ少ないメモリを大切にするため。メモリをソースプログラムを置くために使うと，それだけコンパイラやアセンブラ自体が小さくならないといけない。小さくするためには機能を削らないといけなくなる。機能を削らないと，メモリが少なくてソースプログラムが入る所がなくなってしまうというわけ。

ところでテープでは，この“外から読みながら外に出す”ができない。シーケンシャルアクセス* しかできないから。ソースプログラムもコンパイラが作り出すマシン語プログラムもファイルになっているよね。このファイルへ書くことやファイルから読むことが，テープでは，はじめから順にしかできない。だからテープでは，こっちのファイルを読みかけでやめて，もう一つのファイルに書きたいっていうのはダメなんだ。ところがディスクではこれができる。“外から読みながら外に書く”ができて，だからメモリ上にソースプログラムも，アセンブラやコンパイラによって作られるマシン語プログラムも置かないで，アセンブルやコンパイルができてしまうんだ。

また，ソースプログラムの整理を考えると，そのために必要な機能は，「今どんなソースプログラムのファイルを持っているかわかる」「その内の適当な一つを選んで，その中身（何んのプログラムか）が読めるようになっている」とか「不要なファイルは消せる」などだ。しかもこれらの機能ができるだけ速くなされないといけない。

上のどれも，一応はテープでもできる（KILLやFILESなど)けれど，たとえば，「ファイルが10個あって，その中身を調べながら，ある目的に不必要な３個を捨てる」なんて作業は，健康のために絶対やってはいけない。ディスクとOSがあれば，こんなことは朝飯前にできる。FLEXもOS-9もこうしたファイルに関する便利なコマンドをちゃんと用意しているし，ユーザーが自分に適したコマンドを作りやすいようにもできているので，ソースプログラムの整理が簡単にすぐにできて，いま自分がどんなプログラムの蓄えを持っているか把握していられるし，いつでも取り出して利用できる。OSっていうのは，そういうもののことなんだ。いってみればツールのかたまり。

OS上で走るアセンブラやコンパイラもまたすごい。たとえばFLEXのマクロアセンブラは，ラベルをつけられる。これは，たとえば飛び先番地などに名前（ラベル）を付けておいて

BRA　　名前

とすれば，あとはアセンブラが番地の計算をしてくれる，という機能。ほかにマクロ機能，条件付きアセンブル機能など，ややこしいアセンブリを少しでも書きやすく読みやすくするための機能がいっぱい盛り込まれている。

この“読みやすく”っていうことも，プログラムの開発では，とても大切なことなんだ。さっきも言ったけれど，プログラム作りは積み上げ作業だし，前に使ってもう絶対役に立たないプログラムは少なくて，改良や付け加えをして使うという作業は必ずある。そんなとき，自分で書いたプログラムが後で読むと全然わからない（よくあることだけれど)，あるいは解読に手こずる，というのでは非能率。マクロアセンブラなどでは，読みやすく理解しやすいアセンブルプログラムが書けるようにできている。

まだ予算があるあなた

ディスクとOSの基本システム（たいがいアセンブラはついてくる）を買ってもまだ予算があまって困るというあなたには，デバッガというツールをおすすめする。FLEXにもOS-9にも用意されている。

BASICでプログラムを作るとき，みんなTRONとかTROFFとかのトレース機能のお世話になるでしょ？　このデバッガはマシン語のそれができる。マシン語を１ステップずつ実行して，そのときの各レジスタの値の表示などもしてくれる。しかも暴走しない！　ほかにもブレークポイントが設定できるとか，いろいろな機能があって，マシン語プログラムのバグ取りに大いに役立つ。

もっと予算がある人なら，OS上の高い高い高級言語でマシン語プログラム作りはいかがですか。ゲームやシステムを作るなら，OS-9のCやPascalとか，FLEXのmicro-CやPascalをおすすめしますよ。FORTHもFLEXとOS-9のどちらにもあって，これはロボットの制御や，音楽，計測などをやってみたい人に向いている。

また，よくコンパイラで作られたマシン語プログラムは，アセンブラで作ったマシン語プログラムより遅いしメモリを食う，というけれど，習熟していない人なら，アセンブラを使うより気に入った言語のコンパイラを使った方が，結構コンパクトで速いプログラムが作れるかもしれない。というのも，6809というのはとてもよくできたCPUで，人間にとってみてもZ80なんかと比べて，アセンブリでプログラムが作りやすい（論理的な命令体系になっている）し，コンパイラにとってもマシン語プログラムが作りやすくできている（高級言語志向なんて呼ばれている)。だから，6809のアセンブラに慣れた人（すぐに慣れるよ）なら，効率的で速いマシン語プログラムを特殊なテクニックなしで書けるし，アセンブラをやりたくない人でも，高級言語のコンパイラで，アセンブラで書いたのに劣らないマシン語プログラムが作れる。

ツールだけ揃えても

結局はこれです。COMSOLやＫは，ハンドアセンブルやミニアセンブラはとっても面倒くさいんだなってわかった人が買うべきだし，ディスクも，カセットでコンパイルする感覚に耐えられなくなった人が買うべきだと思う。結局，自分の習熟度にあったツールを使うのが一番。

でもその一方で，良いプログラムを作るには良いツールが不可欠だというのも本当で，やっぱりカセットだけでマシン語ゲームを作るのは無理なのです。

いろいろ苦労はあるけど，マシン語を勉強して一番報われるのは，FMユーザーに違いない。これは絶対だよね。マシン語を勉強するなら6809，68000など68系に限る，というのが心ある人の一致した意見，といえるから。

* シーケンシャルアクセスおよびランダムアクセスについては，Oh! FM第１号の「F-BASIC中級入門」および第３号の「ベーシックBASIC」に詳しく書かれているので参照してほしい。

マシン語特集 part II

2

初心者向けプログラム開発話
例題：外字登録エディタ

本GO 誠司

マシン語のプログラムを作ってみよう，と思っても，どう手をつけたらよいかわからない。そんな人のために「外字登録エディタ」プログラムを例にとって，開発手順やマシン語プログラム開発にあたっての注意事項などを述べてみましょう。

イントロ

マシン語のプログラムだけではなく，どんなプログラムを作るときにも言えることですが，目標というか，しっかりした目的などを持って取り組むことが大切です。もちろん初めのうちは，「この命令はどんなことをするのだろうか……」と，いろいろ試してみることが必要でしょう。しかし，それらの命令はあくまでもプログラムを作るときに必要な道具であって，プログラムそのものではないことを忘れないでください。つまり，「この命令でこんなプログラムができる」のではなく，「こんなプログラムを作ろう。それにはこの命令が使える」ということなのです。ですから，どんなことをやらせたいプログラムを作るかを常に念頭に置いて，プログラムを作るようにしなければならないのです。

さて，何を作ろうか？

それでは最大の難関である何をやるプログラムを作るか，から入りましょう。

私は以前よりFMシリーズの漢字表示に感激し，ちょっとした文章や，ひいてはレポートなどの作成にも使用してきました。なにしろ，パソコンでの漢字表示のパイオニアはFMシリーズなんだもんねぇ，皆さん!! しかし，JIS第1水準では表示できない漢字や記号があります。何とかBASICレベルの漢字表示コマンドで，自分の作った漢字や記号を表示できないかと思ったものです。が，そのためにはF-BASICの内容がわからなければダメなのと，多忙とが重なりしばらく忘れていました。そうこうしているうちに，FM-8にもちょっとナマイキな弟分の7や，ほんとにエライ兄貴分の11が登場し，また，F-BASICの解析なども進んできました。特に，11は漢字関係の機能が強化され，しかも私のやりたかった外字登録機能もあるではありませんか！ これはなんとしても7，8でも外字登録機能がほしい！ というのが，今回のプログラム作成テーマになったわけなのです。

さあ，作るぞ！

テーマが決ったら，まずシステム設計をしましょう。今回のプログラムは，以下のシステムが必要です。

1）FM-7 or 8本体
2）ディスプレイ（カラー，グリーンいずれも可）
3）外部記憶装置（カセットテープ，フロッピーディスク，またはバブルカセット）ただしバブルカセットの場合，ファイルディスクリプタの変更が必要

基本的には漢字ROMが載っていることを前提としていますが，なくても可能です。

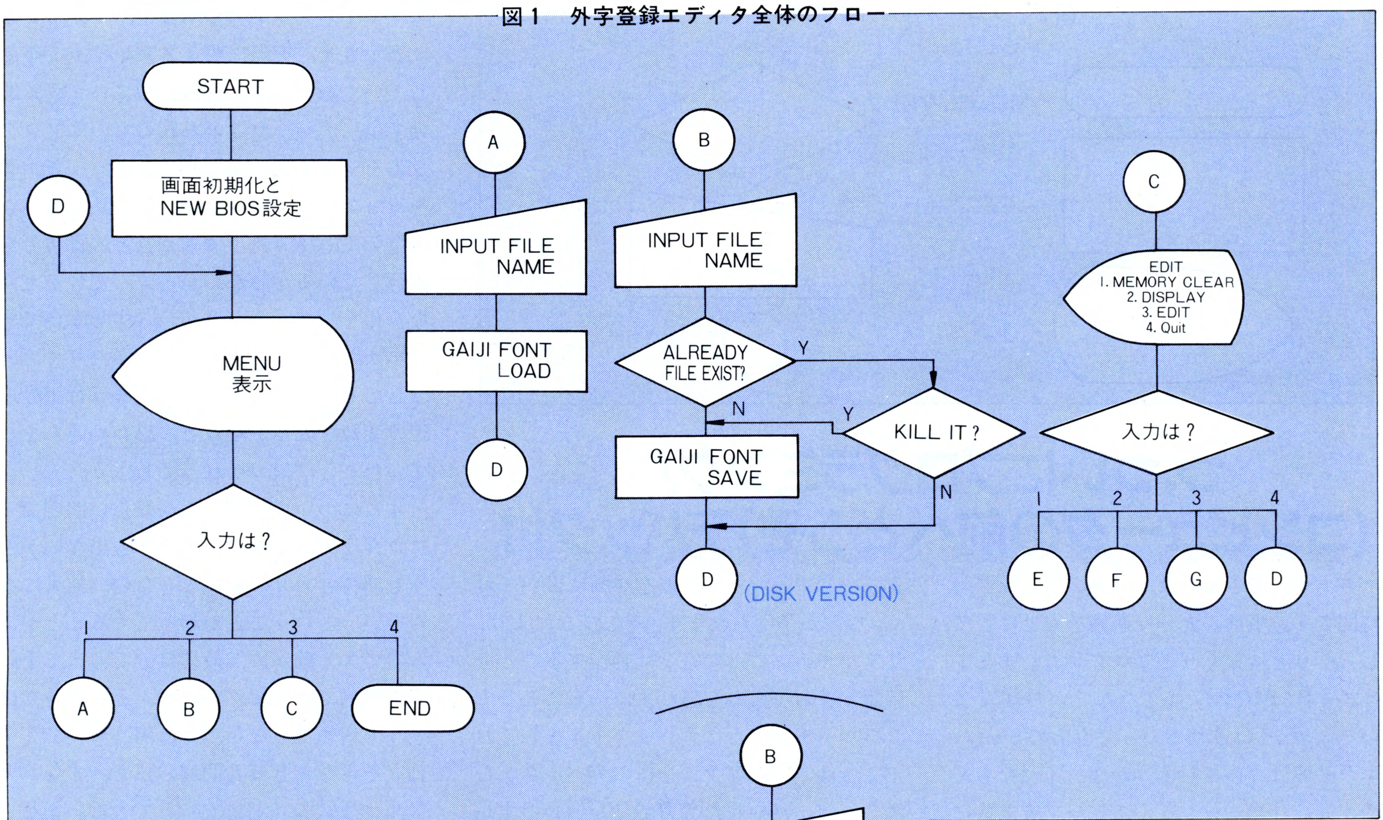
次にプログラムそのものの仕様として，

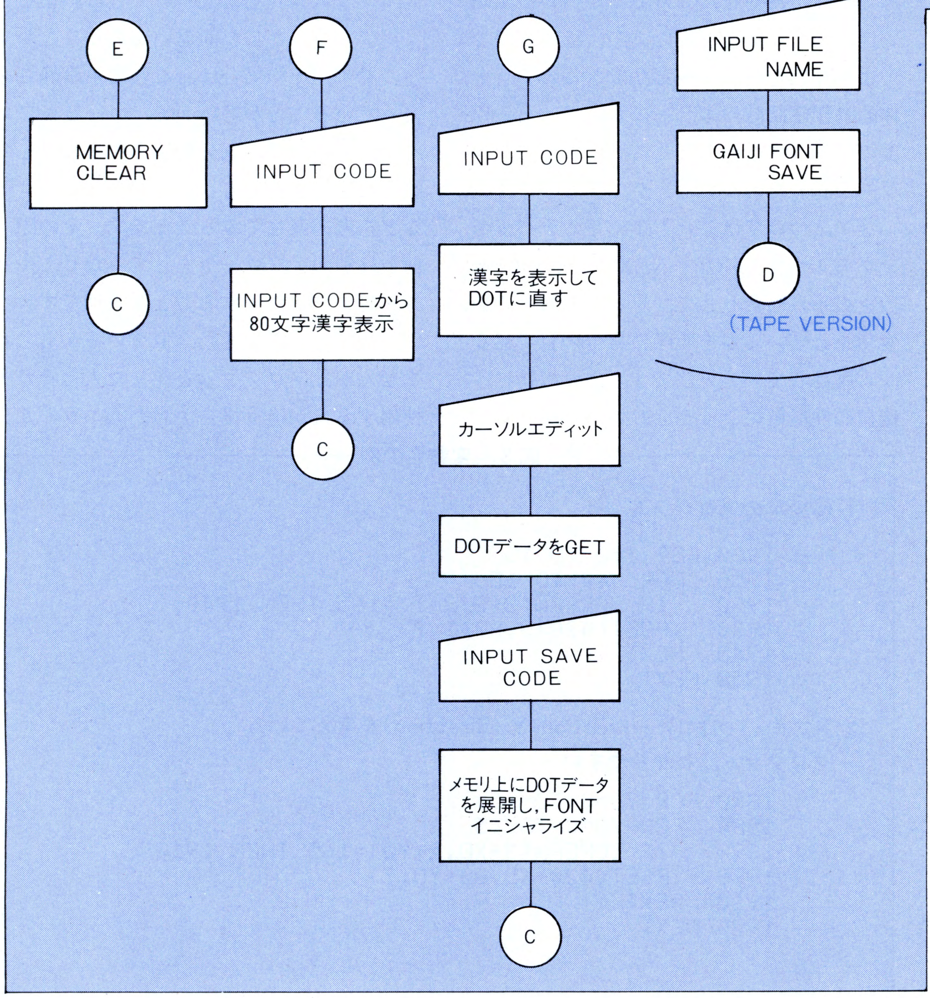
1）画面は見やすく40×25モード
2）16×16の漢字フォントを左側に大きく表示し，その右にMENUと実物大を表示
3）画面の下に2行分の漢字表示部分を残すこと
4）漢字のエディットはカーソルキー4個と，INS，DELキーを使用しCRで終了とする
5）MENUの選択は数字で行う
6）エディットする漢字はROM上のものでも外字でも可とし，任意の外字コード名で登録できるようにする
7）外字のデータは外部にセーブ，外部からロードできるようにする

だいたい以上がプログラムの仕様となりますが，実際にはプログラミング中で変更が必要になることがありますので，大きな目安ぐらいということにしておきましょう。

次に，フローチャートを書くことにしましょう。フローチャートを書くことは，プログラムの全体的な動きを知る上で，とても大切なものだと思います。「そんなもの頭の中でできるからカット，カット」とおっしゃる方も，もう一度考え直してくださいな。皆さんもご存知のとおり，コンピュータは偉大な白痴であります。これしろ，あれしろ，と命令を与えてやればイヤな顔せず，文句も言わず黙々と仕事をやり続けますよね。たとえその内容が，正しくないときでも。

図1　外字登録エディタ全体のフロー



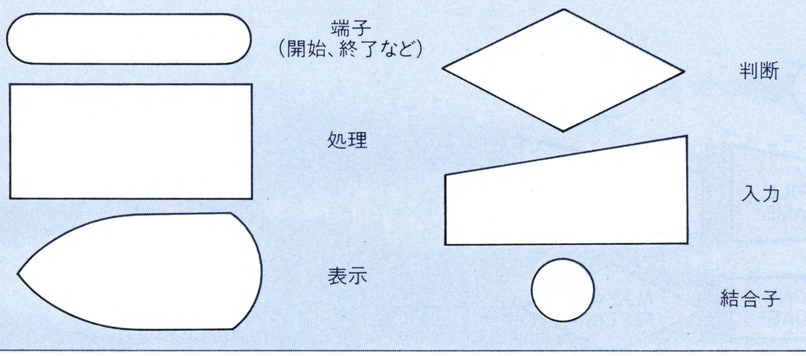


ここが大切なところで，こちらでさせようとしている内容に間違いがあっても，コンピュータは黙って仕事をするだけです(エラー処理などとは別ですョ，根本的な人間的誤ちのことです)。ですから，何をさせるか，そしてその内容を正しくコンピュータに教える（プログラミングする）ために，自分自身がまずしっかりと内容を把握しなくてはならないのです。

図1を見てください。これが今回作ろうとしている「外字登録エディタ」の全体的フローです。読み方は，STARTから始まって，上から下へと進みます。おのおのの記号の示す意味は**図2**のとおりで，内容も名称のとおりです。実際の仕事は，その記号の中に書き込みます。

なるべくならば細分化した方がいいのですが，まずは大まかにでも書くようにしてみてください。その次にその内容をさらに詳しく別に書き出したりすることをおすすめします。この作業を事前にやるかやらないかで，後々のデバッグ（虫取り）の能率が違ってきます。特にマシン語のプログラムを組むときのフローチャートは，1記号に1ステップ，とやった方がいいことは，言うまでもありません（経験者は語る！）

図2　フローチャートの記号



つ,ついにプログラミング!(コンピュータの前へどうぞ)デバッグ付

さあやってきました，プログラミングの時間です。事前の充分な準備をもとにして，じっくりとゆっくりとプログラムしましょう！　BASICのプログラムでいちばん多いミスは，やはり何といってもあのSyntaxエラーではないでしょうか？　イヤなもんですョ，「ピーッ」なんてビープ音と共に「Syntax Error in ××」で止ってしまうのは……。ですから，つまらない打ち込みミスをしないように心がけましょう。そして，まずはBASIC部分だけで一応動くようになるまで，デバッグをします。ではその方法はいかに!?　以下にその要点を列挙しましょう。

1）変数の一覧表を作る

2）サブルーチンごとに動作を確認する（特に変数の引き渡しなど）

3）プログラムの随所で変数を表示させて，実際に期待した値になっているかを確認する

4）アルゴリズム（人間の考え方）に間違いがないかの確認

5）作ったプログラムを他人に実行してもらう

1）についてはわかりやすい変数名を使うようにすると，デバッグが楽です。ただし，気をつけるのは予約語を含まないようにすることです。ここらのことは「F-BASIC文法書」をよく読んでください（**図3**）。

2）についてはたとえばそのサブルーチンをコールする際の変数と同一名のサブルーチン名にするとか，サブルーチンのリマークのところへ，このルーチンをコールする際のフォーマットなどを詳細に書いておく，などが挙げられます。そしてなにより，サブルーチンごとに完全に動作することを確認しておくことを忘れずにやることです。サブルーチンごとにデバッグをすましておくことは，プログラム全体のデバッグから考えれば，たいした労力ではないのですから。しかも，変数などに実際にはあり得ないような値を入れてみたりできるので，全体的動作確認時のエラーにおいて，ある程度の予測（すなわちどこらへんでどんなエラーが起ったのかなど）ができるのです。

3）についてはこれもよくやるデバッギングの1つで，FOR～NEXTループ内で変数を表示させたりして，思ったとおりの値，変化をしているかを実際に自分の目で見ながら確認していく方法です。これは特に，機械語開発時によくやられています。レジスタやインデックスの内容を，1命令ごとに表示させ，実際にディスプレイ上で確認していくわけです。機械語のプログラム開発では，デバッガといわれるものにプログラムをシミュレートさせて，デバッギングを行っています。BASICにも似たような命令でTRONがありますが，どの行番号を実行しているかしかわからず，変数がどんな変化をしているかはわからずじまいです。ですから，ここぞと思われるあたりや，どうしてもエラーが発生してしまう行番号近辺で変数を表示させると，思わぬ誤ちをしていたことに気づかれるでしょう。

4）についてはなかなか難しい問題です。プログラムは完成し，エラーも出ないがどうも期待どおりの値にならない，結果にならない，ということが起ってくることがあるのです。確かに，最初自分で考えて手計算の数例ではうまくいくが，それ以外ではどうも信頼できない，では困りものですよね。そんなときは人間側の誤ち，すなわちアルゴリズム（考え方，とでも言いましょう）にちょっとした間違いがあることが多いのです。その場合は，たいてい冷静さを失っていることが多いので，コーヒーブレイクにするなり，気分転換をするなりして頭を冷やしてからもう1度，全体のフローなどを考え直してみることです。その際，前とは別のアプローチをしてみたり，やり方を変えてみたりすることが大切です。

5）については4）と似ているかもしれませんが，プログラムを作った人とそれを使用する人が違う場合，思わぬエラーが出

図3　変数名の例

①IJ少年のプログラム

```
1900 FOR I=0 TO 15
1910  FOR J=0 TO 15
1920   IF SCREEN(3+J,2+I)=165 THEN 1940
1930   PSET(420+J,80+I,7)
1940  NEXT
1950 NEXT
```

②X方向・Y方向(X−direction, Y−direction)を考えているプログラム。わかりやすい

```
1870 FOR YD=0 TO 15
1880  FOR XD=0 TO 15
1890   IF SCREEN(3+XD,2+YD)=165 THEN 1910
1900   PSET(420+XD,80+YD,7)
1910  NEXT
1920 NEXT
```

ることがあります。この「思わぬエラー」というのが曲者で，プログラム製作者は自分でデバッグしているときには，どうしても人情として無理をしないことが多いようです。というか，エラーを出さないようにデバッグしがちなのです。ですからあなたの周辺の友人に実際にプログラムを実行してもらい，あなた自身では思いもよらなかった行動で，思わぬエラーを発見してもらうのです。これは特に，ゲームを作ったときに効果絶大です。思わぬバグや必要法をみつけるのは，私の場合もご多分にもれず多くの場合私の友人たちでした。

マシン語の場合

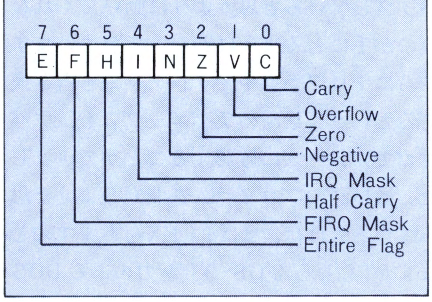
機械語開発も基本的には前述と同様の手順に従って行います。そうまず，「何をしたいか？」，次に「フローチャート」「プログラミング」，「デバッグ」の繰り返しの末に「完成」となるわけです。

しかし，機械語はBASICに比して低級言語ですから何といってもあの「暴走」という恐ろしい現象が，常に大きな口を開けて待っています（BASICのエラー表示がとても便利なものに見えてくるのはこのときではないでしょうか？）。暴走族にならないようにするための注意点を思いつくままに書いてみましょう。ただしCPUは6809です。

1）コンディションコードから目を離すな！
2）アドレッシングモードに注意せよ！
3）2進数を思い出せ！
4）いきなりランは暴走するぞ！
5）スタック操作は慎重に！

1）については，機械語ではビット単位から2バイト（すなわち1～16ビット長）までの操作が1命令の単位となります。その操作によって生じた結果内容によって変化するのが，コンディションコードレジスタです（**図4**）。このレジスタがどんな状態であるか，どんなふうに変化するかに常に目をやる習慣をつけるようにしましょう。そうすれば，条件ブランチ命令の誤りはなくなるはずです。

図4　CCRの図



2）についての特に多い間違いは，イミディエイトと，エクステンディドダイレクトの間違いではないでしょうか。**図5**のような間違いを今でもやってしまいます。アセンブラでたった1文字違いなのですが。気をつけましょう。

3）については符号なし2進，符号付き2進，というような表現でよく耳にすると思いますが，両者の決定的違いは最上位ビットの重みです。前者ではMSB（最上位ビット）が＋128で，後者ではMSBが－128となります。すなわち符号なしでは，10000001(2)は，81(16)，＋129(10)で，符号付きでは2進，16進は同じでも－127(10)となるわけです。ですから0～255，－128～127という違いになってくるわけです。このことも思い出し，さらにコンディションコード中のネガティブフラグのことも考慮に入れれば，ヘンテコなミスはなくなるはずです。

4）についてはあまりにも有名でしょうから口うるさく言うのはよしましょう。でも少し言わせてもらえば，小さな確実に動くルーチンを集めて，目的とする大きなプログラムを作るようにしましょう。これは非常に大切なことです。小さなうちに完全なものとしてしまうのです。「大きくなってからかかる病気は重い」と言うように，大きなプログラムのデバッグはそれはもう想像を絶するものがあります。「確実に動く小さなルーチン」，これをお忘れなく。

5）については，これは特に6809だから言えることかもしれません。例を挙げると，PSHS AとSTA，－S，PULS AとLDA，S＋はそれぞれ同じです。またRTSとPULS PCも同じものです。このようにスタック操作が強力かついとも簡単に行えるため，逆にその操作には注意を要するわけです。慣れないうちは気をつけると同時に，ユーザースタックの方を使うといいかもしれません。

エンディング

さて，苦労して作ったプログラムの出来はどうですか？　何度作っても完成のときの喜びはこの上もないものでしょう。充分に使いこなし，完成までの苦労があなたのプログラミングに対する自信になることを祈っています。そして，次なるプログラムへの意欲や改良案のことも忘れずに頑張ってください。

図5　アドレッシングモードの間違いやすい例

```
 4000                         ORG    $4000

             5000   CHECK     EQU    $5000

 4000 8E    5000              LDX    #CHECK  ———— イミディエイト  ⎫
 4003 30    05                LEAX   5,X                          ⎬ '#'1つで大きな違い!!
 4005 BF    5000              STX    CHECK   ———— エクステンディド ⎭

                              END    CHECK

0 ERROR(S) DETECTED

SYMBOL TABLE:

CHECK   5000
```

P.S.FLEXのデバッガについて

私は機械語開発にFLEXのアセンブラを使用していますが, その供給会社であるTSCより6809 Debug Packageというのが発売されています。これは非常に強力なデバッガでコマンドが49種もあり, その中にはインラインアセンブラ, ディスアセンブラ, インタラプトコントロールなどまで入っています。トレースも各レジスタの表示のみならず, サブルーチンコールのネスティングの深さや, 今実行している命令のディスアセンブルされたものまで表示され, わかりやすくできています。また, デバッガ自体は6809のプログラムらしくポジションインディペンデントにできているので, ユーザーのプログラムに合わせてロケーションの移動が可能です。CP/MのDDTを使ったことがありますが, ダンチ! というのが正直な感想です。それに歌い文句がニクイではありませんか。「長所は6809の完全なるシミュレート, 短所は実際の200~300倍ほど遅いこと」, TSCの自信のほどがうかがえるような気がします。実際に使用してみると, 本当に涙が出るほど虫がとれて感激したのでした。6809ではCCRはすぐ変化してしまうので, これでOKだと思ってデバッグしていると思うように動いてくれないのが手にとるようにわかってしまいます。さらにプログラム中でプログラムを書き換えるという「禁断のテクニック」も, デバッガでスムーズに確認することができました。機械語のプログラムを開発しようと思っている人には, 私はFLEXをおすすめしますが(もちろんOS-9も魅力のあるDOSですが……) いかがでしょうか。

今回作成のプログラムの説明

プログラムリスト(**図6**)を見てください。1110行から1220行までが, 画面の初期化, 機械語のメモリ上の展開, NEW BIOSの設定, 配列の宣言などを行っています。1260行から1340行がMENUで, ここで各処理に分れます。

MENUで4を選ぶと1340行を通り抜け, 1380行に行きENDとなります。MENUの1と2は同様な処理なので1420行から1470行で判別して処理していますが, 2のセーブ時のエラー発生を考慮して2500行から2540行でエラー処理をしています。私はディスクで作ったのでエラールーチンを入れましたが, テープでやる人はフローチャート同様, 削除してかまいません。ただし, うっかり大切なフォントデータを消さないように。

言い忘れましたが, 初期設定時に1160行のようにON ERROR GOTO~を入れておかないと, エラー処理しませんからそのつもりで。3のGAIJI EDITで, エディットモードになります。エディット時にも, MENUのように処理内容が選択できます。ここらが1510行から1610行で, 4のQuitで1620行に行きMENUに戻ります。1で, GAIJIのフォントデータ領域をクリアします。ここは初めBASICで記述したのですが, 8Kバイトをクリアするのに FM-7の倍速モードでも15秒ほどかかってしまうので, 機械語化しました。その結果約100倍速くなり, 瞬時に終ります。2のGAIJI PRINTでは, インプットした漢字コードから80文字画面に表示します。外字以外の普通の漢字も, もちろん表示できます。

注) 外字は$5000~$507F, $5100~$517Fと$5080~$50FF, $5180~$51FFが同じ表示ですが, これはJISコードとJEFコードをどうもBASIC内で処理しているらしいのです。

3のGAIJI EDITが今回の核ともいうべきところで, またいちばん苦労した点でもあります。特にカーソルコントロールのところがたいへんでした。

まず, 漢字コードを入力します。この機能で既存の漢字をエディットできるわけです。もちろん, 一度登録した外字の訂正もできます。そしてエディットし終ったら再び漢字コードを要求してきますので, ここで外字のコードにあたる$5000~$507Fまたは$5100~$517Fを入力してください。それ以外ではまったく無意味になってしまうので注意しましょう (プログラム内ではじいてくれますが, $5080~$50FFや$5180~$51FFははじきませんから)。そして実際にデータに変換し, メモリ上にストアしているのが1980行から2030行までです。ここまできたら, 2で確認してください。ほら, ちゃんと表示するでしょう? やったネ! 4でMENUに戻り, MENUでも4でQuitしてプログラムを終えてBASICレベルでPRINT@命令でセーブしたときのコードを表示させてみてください。ちゃんと表示するでしょう!?

こうして作った外字データと1110行から1150行と2580行から2620行があれば, 次からは外字を使えるのです。

これで本当に終り

お疲れさまでした。いろいろと述べてきましたが, やはりコンピュータに触れるのがいちばんだと思います。習うより慣れろ! この言葉, 好きですネ。

図6 外字登録エディタ(FM-7,8用)

○このプログラムは, 個人で利用するほかは著作権法上 無断複製を禁じられています。
COPY RIGHT © 1983. S.HONGO

```
1000 '*********************************************
1010 '*
1020 '* GAIJI TOUROKU EDITOR
1030 '*
1040 '* GAIJI CODE=&H5000-&H507F [=&H5080-&H50FF]
1050 '*         and &H5100-&H517F [=&H5180-&H51FF]
1060 '*
1070 '*********************************************
1080 '*
```

```
1090 '* INITIALIZE ROUTINE
1100 '*
1110 WIDTH 40,25:CONSOLE 0,25,0,0:COLOR 7,0:CLEAR 300,&H3FFF
1120 FOR MDATA=&HFC00 TO &HFC6D
1130  READ MDATA$:POKE MDATA,VAL("&H"+MDATA$)
1140 NEXT
1150 EXEC &HFC00
1160 ON ERROR GOTO 2500
1170 DIM GAIJI%(15)
1180 LOCATE 3,0:PRINT "++++++ GAIJI TOUROKU EDITOR ++++++"
1190 PRINT SPC(3);"0123456789ABCDEF
1200 GOSUB 2250
1210 LINE (417,78)-(438,97),PSET,7,B
1220 LINE (420,80)-(436,96),PSET,0,BF
1230 '*
1240 '* MAIN MENU
1250 '*
1260 LOCATE 20,3:PRINT"MENU"
1270 LOCATE 21,5:PRINT"1:GAIJI LOAD  "
1280 LOCATE 21,6:PRINT"2:GAIJI SAVE  "
1290 LOCATE 21,7:PRINT"3:GAIJI EDIT"
1300 LOCATE 21,8:PRINT"4:Quit"
1310 INKY$=INPUT$(1)
1320 IF INKY$<"1" OR INKY$>"4" THEN BEEP:GOTO 1310
1330 INKY=VAL(INKY$):LINE(336,(4+INKY)*8)-(528,(4+INKY)*8+7),AND,2,BF
1340 ON INKY GOTO 1420,1420,1510
1350 '*
1360 '* END OF PROGRAM
1370 '*
1380 WIDTH 80,25:END
1390 '*
1400 '* DATA LOAD & SAVE
1410 '*
1420 LOCATE 20,13:PRINT "Input File Name"
1430 LOCATE 20,14:INPUT FILENAM$
1440 IF INKY=1 THEN LOADM FILENAM$ ELSE SAVEM FILENAM$,&H4000,&H5FFF,0
1450 LOCATE 20,13:PRINT SPC(16)
1460 LOCATE 20,14:PRINT SPC(16)
1470 GOTO 1260
1480 '*
1490 '* EDIT MODE
1500 '*
1510 LOCATE 20,3:PRINT "EDIT"
1520 LOCATE 21,5:PRINT "1:MEMORY CLEAR"
1530 LOCATE 21,6:PRINT "2:GAIJI PRINT"
1540 LOCATE 21,7:PRINT "3:GAIJI EDIT"
1550 LOCATE 21,8:PRINT "4:Quit"
1560 INKY$=INPUT$(1)
1570 IF INKY$<"1" OR INKY$>"4" THEN BEEP:GOTO 1560
1580 LOCATE 20,13:PRINT SPC(20)
1590 LOCATE 20,14:PRINT SPC(20)
1600 INKY=VAL(INKY$):LINE(336,(4+INKY)*8)-(560,(4+INKY)*8+7),AND,3,BF
1610 ON INKY GOTO 1640,1690,1800
1620 GOTO 1260
1630 '*
1640 '* MEMORY CLEAR
1650 '*
1660 EXEC &HFC5B
1670 GOTO 1510
1680 '*
1690 '* GAIJI PRINT
1700 '*
1710 LOCATE 20,13:COLOR 4:INPUT "Input Code &H";CODE$:COLOR 7
1720 CODE=VAL("&H"+CODE$)
1730 LOCATE 20,13:PRINT "Display From &H";CODE$
1740 LOCATE 30,14:PRINT "to &H";HEX$(CODE+80)
1750 LINE(0,160)-(639,199),PRESET,,BF
1760 PRINT@(0,160),CODE   ;:FOR KAN=CODE+ 1 TO CODE+39:PRINT@KAN;:NEXT
1770 PRINT@(0,180),CODE+40;:FOR KAN=CODE+41 TO CODE+79:PRINT@KAN;:NEXT
1780 GOTO 1510
```

```
1790 ´*
1800 ´* GAIJI EDIT
1810 ´*
1820 LOCATE 20,13:COLOR 4:INPUT "Input Code &H";CODE$:COLOR 7
1830 CODE=VAL("&H"+CODE$):PRINT@(420,80),CODE
1840 GOSUB 2400
1850 GOSUB 2100
1860 LINE(X0-1,Y0-1)-(X0+16,Y0+8),PSET,0,B
1870 FOR YD=0 TO 15
1880  FOR XD=0 TO 15
1890   IF SCREEN(3+XD,2+YD)=165 THEN 1910
1900   PSET(420+XD,80+YD,7)
1910  NEXT
1920 NEXT
1930 GET@(420,80)-(435,95),GAIJI%,G
1940 LOCATE 20,13:INPUT "Input Save &H";CODE$
1950 CODE=VAL("&H"+CODE$)
1960 IF CODE<&H5000 OR CODE>&H51FF THEN 1940
1970 IF CODE>&H50FF THEN CODE=CODE-&H80
1980 CODE=(CODE-&H5000)*32+&H4000
1990 FOR STOR=0 TO 15
2000  HD$=RIGHT$("000"+HEX$(GAIJI%(STOR)),4)
2010  POKE CODE+STOR*2,VAL("&H"+LEFT$(HD$,2))
2020  POKE CODE+STOR*2+1,VAL("&H"+RIGHT$(HD$,2))
2030 NEXT
2040 GOSUB 2250
2050 LINE(420,80)-(436,96),PSET,0,BF
2060 GOTO 1510
2070 ´*
2080 ´* KEY IN & CURSOR ROUTINE
2090 ´*
2100 X0=48:Y0=16:X1=48:Y1=16
2110 LINE(X0-1,Y0-1)-(X0+16,Y0+8),PSET,0,B
2120 LINE(X1-1,Y1-1)-(X1+16,Y1+8),PSET,4,B
2130 INKY$=INKEY$
2140 IF INKY$="" THEN 2130
2150 IF INKY$=CHR$(&H7F) THEN LOCATE X1/16,Y1/8:COLOR 7:PRINT "・":PRESET(417+X1/
16,78+Y1/8)
2160 IF INKY$=CHR$(&H12) THEN LOCATE X1/16,Y1/8:COLOR 7:PRINT "●":PSET(417+X1/16
,78+Y1/8,7)
2170 X0=X1:Y0=Y1
2180 IF INKY$=CHR$(&H1C) THEN IF X1=288 THEN 2110 ELSE X1=X1+16
2190 IF INKY$=CHR$(&H1D) THEN IF X1= 48 THEN 2110 ELSE X1=X1-16
2200 IF INKY$=CHR$(&H1E) THEN IF Y1= 16 THEN 2110 ELSE Y1=Y1-8
2210 IF INKY$=CHR$(&H1F) THEN IF Y1=136 THEN 2110 ELSE Y1=Y1+8
2220 IF INKY$=CHR$(&H0D) THEN RETURN
2230 GOTO 2110
2240 ´*
2250 ´* INIT FONT FLAME
2260 ´*
2270 LOCATE 0,2
2280 FOR INF=1 TO 16
2290  PRINT SPC(2);HEX$(INF-1);STRING$(16,"・")
2300 NEXT
2310 RETURN
2320 ´*
2330 ´* ERASE DOT
2340 ´*
2350 LINE(320,10)-(639,77),PRESET,,BF
2360 RETURN
2370 ´*
2380 ´* DOT to FONT CONVERT
2390 ´*
2400 FOR XD=0 TO 15
2410  FOR YD=0 TO 15
2420   LOCATE 3+XD,2+YD
2430   IF POINT(420+XD,80+YD) THEN PRINT "●" ELSE PRINT "・"
2440  NEXT
2450 NEXT
2460 RETURN
```

```
2470 '*
2480 '* ERROR ROUTINE
2490 '*
2500 IF ERR=63 THEN LOCATE 20,13:COLOR 2:PRINT "File ｶﾞ ｱﾘﾏｾﾝ!!":BEEP:FOR TD=0 T
O 1000:NEXT:GOTO 2540
2510 IF ERR=64 THEN LOCATE 20,13:COLOR 2:PRINT "File ﾊ ｽﾃﾞﾆｱﾘ!!":BEEP ELSE RESUM
E NEXT
2520 COLOR 4:LOCATE 20,14:PRINT "Kill or Redo ?":COLOR 7
2530 INKY$=INPUT$(1):IF INKY$="K" OR IBKY$="k" THEN KILL FILENAM$:LOCATE 20,13:P
RINT "NOW SAVING    ":LOCATE 20,14:PRINT FILENAM$:RESUME 1440
2540 COLOR 7:RESUME 1420
2550 '*
2560 '* MACHINE LANGUAGE DATA
2570 '*
2580 DATA 8E,FC,06,9F,DF,39,34,7F,86,16,A1,84,27,06,35,7F,6E,9F,FB,FA,EC,04,81
2590 DATA 50,25,F4,27,0A,81,51,27,02,22,EC,CB,80,E7,05,EE,02,10,8E,00,00,C6,10
2600 DATA 10,AF,C1,5A,26,FA,C6,10,34,04,EC,04,86,20,3D,EE,02,34,10,8E,40,00,10
2610 DATA AE,8B,30,02,10,AF,C1,6A,62,26,F4,35,10,35,04,E7,01,35,7F,1C,FE,39,34
2620 DATA 30,8E,40,00,10,8E,00,00,10,AF,81,8C,60,00,26,F8,35,B0
```

図7　図6のデータ部分の解説

```
                     **********************************
                     *
                     * GAIJI AND KANJI OUTPUT ROUTINE
                     *
                     **********************************        コメント

                     *
                     * FIXED VALUES
                     *

             FBFA    BIOS     EQU    $FBFA   ―― BIOS間接コールアドレス

             0001    RCBSTA   EQU    $0001
             0002    RCBDBA   EQU    $0002    } オフセット定数
             0004    RCBJCD   EQU    $0004

             4000    GAIBUF   EQU    $4000   ―― GAIJIデータのトップアドレス

FC00                          ORG    $FC00   ―― $FC00からプログラム作成

                     *
                     * BIOS ENTRY CHANGE
                     *

FC00 8E   FC06       START    LDX    #NEWBIO ―― X＝(NEWBIOのアドレス)
FC03 9F   DF                  STX    $00DF   ―― $00DFにXをストア
FC05 39                       RTS            ―― リターン
        BIOSのエントリアドレス変更ルーチン
                     *
                     * NEW BIOS ENTRY
                     *

FC06 34   7F         NEWBIO   PSHS   U,X,Y,D,DP,CC ― 本来のBIOSと同様にレジスタ退避(システムスタック上へ)
FC08 86   16                  LDA    #22     ―― A＝22(漢字ROMリードリクエストナンバー)
FC0A A1   84                  CMPA   ,X      ―― インデックスレジスタXの示すアドレス内容とAの内容を比較
FC0C 27   06                  BEQ    GAIJIC  ―― 等しいならば，GAIJICへ，さもなければ次の行へ

        BIOSの
        KANJIR       *
        ルーチン     * GO ORIGINAL BIOS
        コールかの   *
        チェック

FC0E 35   7F         GOBIOS   PULS   U,X,Y,D,DP,CC ― さきほど退避したレジスタを復帰
FC10 6E   9F FBFA             JMP    [BIOS]  ―― BIOSの間接コール
```

```
                          *
                          * GAIJI CHECK ENTRY
                          *
                   FC14   GAIJIC  EQU    *
FC14 EC   04                      LDD    RCBJCD,X      ── D=(4+X:5+X)のメモリ内容 ∴JIS漢字コード(2 byte)
FC16 81   50                      CMPA   #$50          ── Aの内容と$50とを比較
FC18 25   F4                      BLO    GOBIOS        ── もしA<$50ならばGOBIOSへ
FC1A 27   0A                      BEQ    GAIJI         ── もしA=$50ならばGAIJIへ
FC1C 81   51                      CMPA   #$51          ── Aの内容と$51とを比較
FC1E 27   02                      BEQ    ADD           ── もしA=$51ならばADDへ
FC20 22   EC                      BHI    GOBIOS        ── もしA>$51ならばGOBIOSへ
FC22 CB   80              ADD     ADDB   #$80          ── B=B+$80
FC24 E7   05                      STB    RCBJCD+1,X    ── Bを5+Xにストア
   (FC14~FC24: GAIJI CODE ($5000~$507F or $5100~$517F)のチェック)

                          *
                          * GAIJI MAIN ROUTINE
                          *
                   FC26   GAIJI   EQU    *
FC26 EE   02                      LDU    RCBDBA,X      ── U=(2+X:3+X)のメモリ内容 ∴漢字バッファアドレス(2byte)
FC28 108E 0000                    LDY    #0            ── Y=0
FC2C C6   10                      LDB    #16           ── B=16
FC2E 10AF C1              G0      STY    ,U++          ── YをUの示すメモリにストア後,U=U+2
FC31 5A                           DECB                 ── B=B-1
FC32 26   FA                      BNE    G0            ── IF B≠0 THEN G0
FC34 C6   10                      LDB    #16           ── B=16
FC36 34   04                      PSHS   B             ── Bレジスタを退避(システムスタック上へ)
FC38 EC   04                      LDD    RCBJCD,X      ── D=(4+X:5+X)のメモリ内容 ∴JIS漢字コード(2byte)
FC3A 86   20                      LDA    #32           ── A=32
FC3C 3D                           MUL                  ── D=A×B ∴B即ちJIS漢字コードの下位バイトを32倍する(漢字1文字で32byte必要なため)
FC3D EE   02                      LDU    RCBDBA,X      ── U=(2+X:3+X)のメモリ内容
FC3F 34   10                      PSHS   X             ── Xレジスタを退避(システムスタック上へ)
FC41 8E   4000                    LDX    #GAIBUF       ── X=$4000
FC44 10AE 8B              G1      LDY    D,X           ── Y=(D+X:D+1+X)のメモリ内容
FC47 30   02                      LEAX   2,X           ── X=X+2
FC49 10AF C1                      STY    ,U++          ── YをUの示すメモリにストア後,U=U+2
FC4C 6A   62                      DEC    2,S           ── (2+S)のメモリ内容,即ちさきほど退避したBレジスタ内容を1減じる
FC4E 26   F4                      BNE    G1            ── IF (2+S)≠0 THEN G1
FC50 35   10                      PULS   X             ── まずXを復帰 ┐ PULS X, B or PULS B, X とやると,内容は正しく復帰しない
FC52 35   04                      PULS   B             ── 次にBを復帰 ┘
FC54 E7   01                      STB    RCBSTA,X      ── Bは0なので,(1+X)にストア ∴正常に終ったことを意味する
FC56 35   7F                      PULS   U,X,Y,D,DP,CC ── NEWBIOで退避したレジスタを復帰
FC58 1C   FE                      ANDCC  #$FE          ── キャリーフラグをクリア ∴これも正常に終ったことを意味する
FC5A 39                           RTS                  ── リターン
   (MAIN ルーチン; FC26~FC32: バッファ(32 byte)のクリア; FC34~FC4E: GAIJIデータの転送; FC50~FC5A: 正常復帰)

                          *
                          * CLEAR MEMORY from $4000 to $5FFF
                          *
FC5B 34   30              CLR     PSHS   X,Y           ── ここで使用するレジスタの退避
FC5D 8E   4000                    LDX    #$4000        ── X=$4000
FC60 108E 0000                    LDY    #$0000        ── Y=0
FC64 10AF 81              C0      STY    ,X++          ── YをXの示すメモリにストア後,X=X+2
FC67 8C   6000                    CMPX   #$6000        ── Xの内容と$6000を比較
FC6A 26   F8                      BNE    C0            ── IF X≠$6000 THEN C0
FC6C 35   B0                      PULS   X,Y,PC        ── 使用レジスタとプログラムカウンタを復帰
   ($4000から$5FFFまでのメモリクリア)

                                  END    START         ── プログラムの終了

0 ERROR(S) DETECTED                                    ── エラーが発生しなかったというリポート

SYMBOL TABLE:                                          ── ラベル(シンボル)名とそのアドレス表

ADD     FC22   BIOS    FBFA   C0      FC64   CLR     FC5B   G0      FC2E
G1      FC44   GAIBUF  4000   GAIJI   FC26   GAIJIC  FC14   GOBIOS  FC0E
NEWBIO  FC06   RCBDBA  0002   RCBJCD  0004   RCBSTA  0001   START   FC00
```

マシン語特集partⅡ

3

6809セルフアセンブラ

桑 原 岳 夫

はじめに

マシン語を使ってプログラムを組んでみたいが，そのための道具がない，という人は多いと思います。6809のマシン語はアドレッシングモードが豊富で，ハンドアセンブルをするには苦労します。アセンブラがないとなかなか大変です。

フロッピディスクを持っている人なら，68系の標準的なDOSであるFLEXを購入すれば，非常に有用なアセンブラが付属してきます（これは，ほとんどモトローラ社の発表している仕様に準拠しており，マクロ機能* なども完備して，6800の命令コードも6809用に展開してくれる優れたものです。ちなみに当アセンブラもこれを用いて開発しました）。これが現在のところでは，最良の開発ツールと言ってもよいでしょう。

そこで，当アセンブラはフロッピディスクのない人を対象にして，カセットベースのアセンブラに限定しています。すべての処理をメインメモリ上で行うため，アセンブルできるサイズもたかだか数キロバイトですが，フロッピディスクを購入するまでのつなぎとして使っていただければ幸いです。

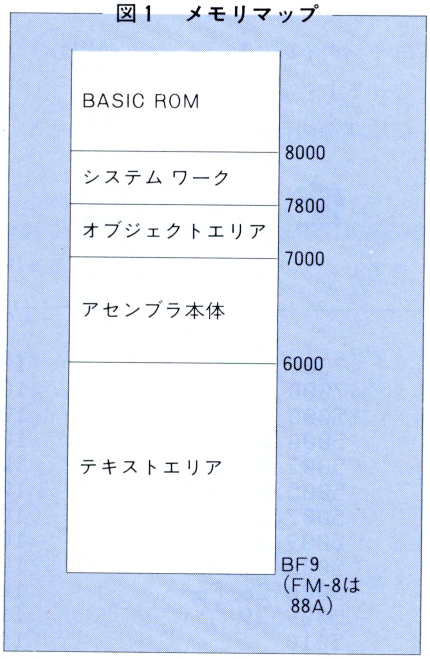
メモリマップ

このシステムのメモリマップを**図1**に示します。当アセンブラはエディタをアセンブラに全面的に依存し，REM文の形でソースを作ります。そのためにソーステキストのLOADおよびSAVEは通常のBASICのプログラムと同様に行ってください。オブジェクトのSAVE，LOADは，$7000番地から行い，固定されています。ただし，ORG疑似命令を持っており，オブジェクトのアドレスの展開は任意の場所に可能です。これはF-BASICのマシン語が任意の場所にオフセット値を付けてロード可能なことを利用したもので，たとえば$6000番地からプログラムをロードしたければ，まず，

SAVEM "ファイル名",&H7000，
終了番地，入口番地

を行ったのちに，

図1　メモリマップ



LOADM "ファイル名",－4096 ⏎
（オフセット値）

でオブジェクトをロードすればよいのです。これで自動的にアドレスが変換されます。この番地でセーブを行えば，次からはこのような作業は不要です。

なお，メモリマップを見てもらえばわかりますが，フロッピディスクについては考慮してありません。

ソーステキスト

ソーステキストは固定フォーマットとなっており，行番号を1000～9999行に限定しています。使い方としては，たとえば1000行から作成するとすれば，

AUTO　1000 ⏎

を実行して，まずREM（REMを使わずに′の形を使う）のためのキーを押し，

1000′ ラベル␣オペコード␣オペランド␣␣

の順に収納していきます。収納効率を上げるために2つ以上のすきま（スペース）があると処理上は無視しますので注意して下さい。

1000′＊

のように，′のあとアスタリスク＊があれば，その行は全てコメントと判断して，処理の上ではスキップします。オペランドの後ろにコメントを置くことはかまいませんが，ソースコードのエリアがせまくなるために，コンパイルするときには省いた方が賢明です。

なお，テキストの例と内部コードの対応を**リスト1**に示します。

＊プログラム作成にあたり何度も使うルーチンを，あらかじめ一つのマクロ命令として定義することにより，記述するときにこのマクロ命令を書けば，アセンブルする際に自動的にマシン語のいくつかの命令の集まりに変換されるという機能。

●マシン語特集II

○ここで扱われるプログラムは，個人で利用するほかは著作権法上 無断複製を禁じられています。 COPY RIGHT © 1983 T. KUWABARA

リスト1

```
1000 ´ ORG $1000
1010 ´*
1020 ´MAIN LDX # $8FCB
1030 ´ END
```

```
088A 08 9B 03 E8 3A 8D 20 4F
0892 52 47 20 24 31 30 30 30
089A 00 08 A3 03 F2 3A 8D 2A
08A2 00 08 BA 03 FC 3A 8D 4D
08AA 41 49 4E 20 4C 44 58 20
08B2 23 20 24 38 46 43 42 00
08BA 08 C5 04 06 3A 8D 20 45
08C2 4E 44 00 00 00 00 00 00
```

（網かけ）アドレスポインタ（2バイト）

（白枠）行番号（2バイト）

ソースフォーマット

仕様としては，少し異なってはいますがモトローラ仕様のサブセットとなっており（詳しくは参考文献(1)を参照のこと），最小限の構成となっています。まず使用できるものは，英数字およびいくつかの記号であり，ラベルについては，6文字以内なら特に制限はありません。オペコードは6809の命令と疑似命令から成り立っています。使用できる疑似命令は，以下の7つです。

① ORG

この命令で実アドレスを指定し，デフォルト（暗黙の指定）は＄7000番地です。BASICと共存するにはここが良いと思います。

② EQU

この命令は，ラベルに数値を割り当てるのに使い，使用するラベルで数値を使いたいところに使用します。

③ RMB

この命令により，指定された量のメモリ領域を空けます。

④ FCB

定数をバイト単位で指定します。複数バイトの指定もカンマで区切って可能となります。

⑤ FDB

定数をワード（2バイト）単位で指定します。そのほかは，FCBと同じです。

⑥ FCC

′と′ではさまれたASCII文字列を定数として展開する命令です。

⑦ END

テキスト終了の印です。

オペランドは，ラベル，レジスタ，式，セパレータ，定数などが置かれます。

16進数は，はじめに＄記号を置いてください。文字は′′でくくってください。ラベルに対するオフセット指定も可能です。なお，アスタリスクはカレントアドレスを示しています。

アドレッシングモードは，

①エクステンド………無指定
②ダイレクト…………＜で指定
③イミディエイト……＃で指定
④インデックス………PCRの場合はラベルが使える
⑤インダイレクト……〔 〕で囲む

となります。

なお，実際の使用例を**リスト2**に示します。

使　用　法

当アセンブラでは，フラグに値を書き込むことにより，

エラーのみの出力か，リスト出力かの選択

を指定します。フラグは＄0番地に0を加えることによりエラーリストのみ，1を加えることにより全リスト出力を指定します。つまり，使用するときは，

```
POKE␣0, :EXEC⏎
       ↑
  0：エラーリスト
  1：全リスト
```

のように，マルチステートメントを使い，オプション指定を行ってください。

お わ り に

実際に機械語のプログラムを開発するときは，アセンブラ，ディスアセンブラ，デバッガなどがそろっていなければ，なかなか苦労します。カセットベースである点にも問題があり，フロッピディスクドライブが大変安くなりつつある現在，あくまでつなぎとして使っていただければ，うれしく思います。

参考文献

(1)6809ハンドブック，加瀬清，アスキー出版
(2)6800→6809クロスアセンブラ，磯貝信男，I/O 56年12月号，工学社
(3)6800 1パスアセンブラ，鈴木優一，RAM 53年7月号，廣済堂出版

リスト2

```
7000              1000 *  REIDAI
7000              1010          ORG   $5000
5000              1020 SPACE    EQU   $20
5000 86 20        1030 MAIN     LDA   #SPACE
5002 8E 0F 0F     1040          LDX   #4095
5005 A6 80        1050 LOOP     LDA   ,X+
5007 26 02        1060          BNE   LO01
5009 1F 89        1070          TFR   A,B
500B E7 A0        1080 LO01     STB   ,Y+
500D 26 F6        1090          BNE   LOOP
500F 39           1100          RTS
5010              1110          RMB   1
5011 4F 48        1120 TABLE    FCC   ´OH´
5013 FC 0A        1130          FCB   $FC,10
5015 FC 01        1140          END
```

リスト3（for FM-7,8）

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
6000	34	7F	C6	6F	1F	9B	F6	00	:98
6008	00	D7	2C	8E	00	33	EC	81	:31
6010	DD	44	EC	84	DD	42	86	7E	:B4
6018	97	80	8E	60	4C	9F	81	8E	:FF
6020	6F	00	86	0E	8D	06	16	08	:B4
6028	2D	A6	46	39	6F	80	4A	26	:B1
6030	FB	39	81	46	2E	13	81	41	:FE
6038	2C	0A	81	39	2E	0B	81	30	:DA
6040	2D	07	8B	07	80	37	1A	01	:98
6048	39	1C	FE	39	A6	46	33	41	:EC
6050	39	81	5D	27	16	81	2C	27	:28
6058	12	81	2B	27	0E	81	2D	27	:C8
6060	0A	4D	27	07	81	20	27	03	:50
6068	1C	FE	39	1A	01	39	0F	18	:CE
6070	9F	02	6D	84	26	06	30	01	:EF
6078	0C	18	20	F4	1F	89	1F	31	:30
6080	9F	04	9E	02	34	02	A6	80	:9F
6088	27	1C	9F	02	9E	04	A1	06	:2D
6090	35	02	27	1A	9E	02	5A	27	:99
6098	04	30	01	20	F9	0C	18	9C	:0E
60A0	00	26	CD	1C	FE	39	35	02	:7D
60A8	9F	02	9E	04	30	1F	30	01	:C3
60B0	9F	04	5A	26	CD	96	18	1A	:B8
60B8	01	39	00	00	4E	4F	50	00	:27
60C0	53	59	4E	43	00	00	00	00	:3D
60C8	00	44	41	41	00	4F	52	43	:AA
60D0	43	00	41	4E	44	43	53	45	:F1
60D8	58	00	45	58	47	00	54	46	:D6
60E0	52	00	4C	45	41	58	4C	45	:0D
60E8	41	59	4C	45	41	53	4C	45	:50
60F0	41	55	50	53	48	53	50	55	:79
60F8	4C	53	50	53	48	55	50	55	:84
6100	4C	55	00	52	54	53	00	41	:DB
6108	42	58	00	52	54	49	00	43	:CC
6110	57	41	49	4D	55	4C	00	00	:CF
6118	53	57	49	00	42	52	41	42	:0A
6120	52	4E	42	48	49	42	4C	53	:54
6128	42	48	53	42	4C	4F	42	4E	:4A
6130	45	42	45	51	42	56	43	42	:3A
6138	56	53	42	50	4C	42	4D	49	:5F
6140	42	47	45	42	4C	54	42	47	:39
6148	54	42	4C	45	42	43	43	42	:31
6150	43	53	42	53	52	4E	45	47	:57
6158	00	00	43	4F	4D	4C	53	52	:D0
6160	00	52	4F	52	41	53	52	41	:1A
6168	53	4C	52	4F	4C	44	45	43	:58
6170	00	49	4E	43	54	53	54	4A	:1F
6178	4D	50	43	4C	52	4C	53	4C	:69
6180	53	55	42	43	4D	50	53	42	:5F
6188	43	00	41	4E	44	42	49	54	:F5
6190	4C	44	00	53	54	00	45	4F	:CB
6198	52	41	44	43	4F	52	00	41	:FC
61A0	44	44	43	4D	50	58	4A	53	:5D
61A8	52	00	4C	44	58	00	53	54	:E1
61B0	58	00	41	44	44	44	4C	44	:F5
61B8	44	00	53	54	44	00	4C	44	:BF
61C0	55	00	53	54	55	00	43	4D	:E1
61C8	50	44	43	4D	50	59	4C	44	:5D
61D0	59	00	53	54	59	00	4C	44	:E9
61D8	53	00	53	54	53	00	43	4D	:DD
61E0	50	55	43	4D	50	53	83	8C	:E7
61E8	8D	8E	8F	C3	CC	CD	CE	CF	:A3
61F0	83	8C	8E	8F	CE	CF	83	8C	:D8
61F8	8E	6F	17	86	11	BD	60	2C	:F4
6200	1F	31	10	9E	0E	A6	26	81	:59
6208	2A	26	01	39	AE	06	8C	4C	:16
6210	42	26	10	0C	1C	9D	80	7E	:3B
6218	62	E2	26	07	0C	1C	9D	80	:B6
6220	7E	62	E2	8E	61	1C	9F	00	:6C
6228	8E	60	BA	86	04	BD	60	6E	:BD
6230	10	24	00	AE	9E	02	6D	82	:71
6238	27	04	C6	02	D7	17	81	10	:72
6240	2B	02	8B	10	8B	10	97	18	:12
6248	9E	04	81	1C	26	0B	A6	06	:1C
6250	30	01	81	43	27	03	0C	1E	:49
6258	39	D6	18	30	01	1F	13	A6	:30
6260	05	BD	60	61	25	1F	C1	3F	:C7
6268	26	EC	D7	19	C6	10	81	32	:8B
6270	27	05	81	33	26	E0	5C	A6	:E8
6278	06	BD	60	61	24	D8	D7	18	:6F
6280	86	02	97	17	39	C1	30	2D	:8D
6288	0B	C1	33	2E	07	8D	3A	24	:1F
6290	C5	16	02	37	96	17	26	11	:F8
6298	C4	FE	C1	1E	26	01	4C	4C	:60
62A0	97	17	81	02	26	02	0C	23	:88
62A8	39	C1	13	26	03	4A	20	F0	:90
62B0	97	17	C4	FC	C1	34	27	EA	:74
62B8	9D	80	81	23	26	98	9D	80	:9C
62C0	BD	60	61	25	91	96	17	20	:01
62C8	D9	1F	31	30	01	A6	05	81	:86
62D0	5B	27	04	81	2C	26	03	1A	:76
62D8	01	39	BD	60	61	24	EC	1C	:E4
62E0	FE	39	8E	61	A2	9F	00	8E	:F5
62E8	61	1C	86	03	BD	60	6E	10	:A1
62F0	24	01	20	81	23	10	2E	00	:27
62F8	B0	81	12	2E	48	E6	06	C1	:66
6300	20	27	03	0C	1E	39	30	01	:DE
6308	1F	13	0C	1D	8B	20	C6	02	:CE
6310	81	20	26	0E	0D	1C	27	03	:28
6318	86	16	5C	97	18	D7	17	0C	:A1
6320	20	39	81	32	26	0D	0D	1C	:68
6328	27	05	86	17	5C	20	EC	86	:B7
6330	8D	20	E8	81	30	2D	02	80	:F5
6338	0C	0D	1C	27	DE	97	19	86	:70
6340	10	C6	04	20	D6	1F	89	C0	:38
6348	13	C1	10	26	02	C6	08	1F	:F9
6350	13	9D	80	81	20	27	24	4D	:69
6358	26	03	0C	1E	39	C1	0E	27	:82
6360	F9	81	41	27	06	81	42	26	:D1
6368	F1	CB	10	CB	40	9D	80	BD	:B1
6370	60	61	24	E6	86	01	D7	18	:41
6378	97	17	39	9D	80	81	3C	26	:E7
6380	0E	9D	80	81	20	26	D3	0C	:D1
6388	22	0C	20	86	02	20	E7	33	:10
6390	5F	BD	62	C9	25	08	0C	20	:A0
6398	CB	70	86	03	20	D8	CB	60	:E7
63A0	D7	18	86	02	97	17	16	01	:3C
63A8	22	1F	89	C0	24	A6	06	81	:DB
63B0	41	27	06	81	42	26	5C	CB	:7E
63B8	40	CB	80	30	01	1F	13	9D	:8B
63C0	80	81	20	27	03	0C	1E	39	:AE
63C8	9D	80	81	23	26	19	0C	23	:2F
63D0	34	04	C4	0F	C1	07	35	04	:0C
63D8	27	EB	9D	80	81	20	26	E5	:DB
63E0	86	02	97	17	D7	18	39	81	:DF
63E8	3C	26	0E	9D	80	81	20	26	:54
63F0	D4	0C	20	0C	22	CB	10	20	:29
63F8	E7	33	5F	BD	62	C9	25	08	:8E
6400	0C	20	CB	30	86	03	20	DA	:AA
6408	CB	20	D7	18	86	02	97	17	:10
6410	16	00	B8	8E	61	E6	9F	00	:42
6418	8E	61	A2	86	04	BD	60	6E	:A6
6420	10	24	03	50	1F	13	34	02	:EF

```
6428  8E 61 E6 4C E6 80 4A 26 :F7
6430  FB 35 02 D7 18 81 10 2D :DF
6438  04 86 11 20 06 81 0A 2D :79
6440  08 86 10 97 18 D7 19 0C :49
6448  17 9D 80 81 20 27 03 0C :0B
6450  1E 39 9D 80 81 23 26 15 :53
6458  C4 8D C1 8D 27 F1 0C 17 :DA
6460  9D 80 81 20 26 E9 0C 20 :F9
6468  0C 17 0C 17 39 81 3C 26 :62
6470  10 CB 10 0C 22 0D 17 26 :63
6478  04 D7 18 20 E3 D7 19 20 :06
6480  DF 33 5F BD 62 C9 25 13 :91
6488  CB 30 96 17 26 04 D7 18 :C1
6490  20 02 D7 19 8B 03 97 17 :4E
6498  0C 20 39 CB 20 0D 17 26 :9A
64A0  04 D7 18 20 02 D7 19 8D :92
64A8  BF 20 20 34 04 8E 64 C7 :F0
64B0  A1 84 27 0E CB 20 30 01 :76
64B8  8C 64 CB 26 F3 35 04 1C :29
64C0  FE 39 32 61 1A 01 39 58 :76
64C8  59 55 53 9D 80 C6 80 81 :E5
64D0  5B 26 04 0C 1F 9D 80 81 :4E
64D8  2C 10 26 00 7B 9D 80 8D :87
64E0  CA 25 15 81 2D 26 2F CB :D2
64E8  02 9D 80 81 2D 26 03 5C :52
64F0  9D 80 8D B7 24 20 20 0A :CF
64F8  9D 80 81 2B 27 4D 33 5F :CF
6500  CB 04 9D 80 0D 1F 27 12 :51
6508  CB 10 81 5D 26 6D 1F 98 :03
6510  84 9D 81 90 26 02 20 63 :DD
6518  9D 80 BD 60 61 24 5C 0D :28
6520  20 27 04 9E 04 1F 13 9E :BD
6528  1B 96 18 81 20 96 17 25 :3C
6530  0D D7 19 D6 1C 9F 1A 81 :29
6538  03 26 02 D7 1A 39 D7 1A :46
6540  D6 1C 9F 1B 81 04 26 02 :59
6548  D7 1B 39 9D 80 81 2B 26 :1A
6550  03 5C 20 AE 33 5F 20 AA :89
6558  81 41 26 04 C6 86 20 0E :66
6560  81 42 26 04 C6 85 20 06 :5E
6568  81 44 26 16 C6 8B 9D 80 :6F
6570  81 2C 26 0A 9D 80 BD 64 :1B
6578  AB 25 D7 0C 1E 39 33 5E :9B
6580  9D 80 8E 00 00 9F 1B C6 :2B
6588  80 D7 06 81 2D 10 27 00 :42
6590  82 BD 60 3A 25 46 0C 20 :70
6598  33 5F 1F 31 9F 04 9D 80 :A2
65A0  81 2C 27 15 BD 60 61 24 :8B
65A8  F5 9E 04 1F 13 0D 1F 27 :1C
65B0  CA C6 9F BD 64 68 16 FF :CD
65B8  6E 1F 31 A6 06 BD 64 AB :36
65C0  10 25 00 E7 C6 8C 0C 17 :91
65C8  0C 25 9D 80 8D 2F 24 AB :D9
65D0  27 04 33 5F 20 03 5C 0C :48
65D8  17 16 FF 26 5F 4D 26 04 :28
65E0  9D 80 20 06 5C BD 67 3A :FD
65E8  27 45 BD 60 32 25 F5 81 :56
65F0  5D 26 07 9D 80 BD 60 61 :25
65F8  25 B3 16 FF 7E 81 50 26 :62
6600  0F 9D 80 81 43 26 09 9D :BC
6608  80 81 52 26 03 1A 01 39 :D0
6610  1C FE 39 5F 9D 80 81 30 :80
6618  27 F9 BD 60 32 24 DB 5C :CA
6620  BD 67 3A 26 F5 03 1B 03 :9A
6628  1C 9E 1B 30 01 9F 1B C1 :81
6630  04 2E C7 9D 80 D6 06 17 :09
6638  FE 71 25 34 C6 8C 8D BD :64
6640  24 B8 26 05 5C 0C 17 20 :A6
6648  02 33 5F 0C 17 9E 1B 9F :0F
6650  06 BD 6A C9 9E 06 9F 1B :54
6658  96 1B 26 06 0D 1C 2A 1F :4F
6660  20 07 43 26 04 0D 1C 2B :E8
6668  16 C1 8D 27 12 16 FF 0B :BD
6670  CB 08 96 1B 27 0C 81 FF :37
6678  27 16 0C 17 5C 0C 17 16 :F5
6680  FE 80 96 1C 2B F4 0D 1F :7B
6688  26 F3 84 F0 26 EF 20 0D :CF
6690  96 1C 2A E6 0D 1F 26 E5 :F9
6698  8A 0F 43 26 E0 C4 60 96 :9C
66A0  1C 84 1F 34 04 AB E0 1F :A1
66A8  89 20 D4 9E 0E 9F 08 1F :EF
66B0  31 30 1F A6 06 81 20 27 :F4
66B8  04 81 5B 26 F4 30 01 9F :CA
66C0  02 D6 06 34 04 9E 44 9F :97
66C8  0E 9E 02 9F 04 9E 0E 9C :99
66D0  42 27 26 9F 06 9E 04 A6 :7C
66D8  06 81 2C 27 0E 30 01 9F :B8
66E0  04 9E 06 A1 06 26 0C 30 :B1
66E8  01 20 E8 9E 06 A6 06 81 :DA
66F0  20 27 0F BD 68 CE 16 FF :5E
66F8  D0 32 61 9E 08 9F 0E 16 :CC
6700  FE 79 1F 31 9F 04 9E 06 :0E
6708  30 01 1F 13 9D 80 81 45 :46
6710  26 E7 9D 80 81 51 26 E1 :03
6718  9D 80 81 55 26 DB BD 67 :18
6720  A3 0D 1E 26 D4 0F 21 0F :07
6728  20 9F 1B 9E 04 1F 13 9E :4C
6730  08 9F 0E 35 04 D7 06 16 :E1
6738  FE F9 48 48 48 48 34 04 :4F
6740  C6 04 48 09 1C 09 1B 5A :B5
6748  26 F8 35 04 9D 80 81 2C :21
6750  39 4F 52 47 52 4D 42 46 :48
6758  44 42 45 51 55 46 43 42 :3C
6760  46 43 43 45 4E 44 67 9E :A8
6768  67 C8 68 26 67 A3 67 CD :FB
6770  67 EF 6E 8D 8E 67 66 9F :4B
6778  00 8E 67 51 86 03 BD 60 :EC
6780  6E 24 18 1F 13 8E 67 66 :37
6788  48 6E 96 9D 80 81 20 26 :30
6790  08 BD 68 DB 24 03 9E 06 :D3
6798  39 32 62 0C 1E 39 8D EB :A8
67A0  9F 0A 39 1F 31 A6 07 81 :60
67A8  2D 27 05 8D DE 0C 21 39 :2A
67B0  9D 80 81 20 26 E5 9D 80 :E6
67B8  0F 06 0F 07 C6 04 BD 69 :1B
67C0  18 24 D8 BD 6A 9D 20 E5 :DD
67C8  8D C1 0C 26 39 0C 24 9D :86
67D0  80 81 20 26 C6 8D 3D C6 :9D
67D8  02 BD 68 DD 24 BD 8D 3B :AD
67E0  4D 26 01 39 1F 31 A6 06 :A9
67E8  81 20 26 EB 39 20 AC 0C :C3
67F0  24 9D 80 81 20 27 F8 81 :82
67F8  27 26 A0 8D 17 9D 80 81 :2F
6800  27 27 09 4D 27 E7 97 07 :50
6808  8D 11 20 F1 9D 80 BD 60 :E9
6810  61 24 DA 39 10 9E 0A 10 :60
6818  9F 04 39 9E 04 D6 07 E7 :42
6820  80 9F 04 0C 17 39 0C 24 :AF
6828  9D 80 81 20 10 26 FF 6B :5E
6830  17 FF E1 BD 68 DB 10 24 :2B
6838  FF 61 8D 0D 4D 26 01 39 :A7
6840  1F 31 A6 06 81 2C 26 EB :BA
6848  39 9E 04 DC 06 ED 81 9F :CA
6850  04 0C 17 0C 17 39 8E 78 :89
6858  00 9F 10 9F 12 9F 14 8E :A1
6860  70 00 9F 0A 9E 44 9F 0E :A8
6868  9E 0E 30 84 10 27 00 CD :64
6870  1F 13 9E 12 C6 07 9D 80 :CC
```

```
6878  81 20 27 0B 25 05 A7 80 :24
6880  5A 26 F3 0C 1E 20 18 C1 :96
6888  07 27 11 5A 27 04 6F 80 :B3
6890  20 F9 10 9E 0A 10 AF 81 :11
6898  9F 12 9F 14 BD 61 F8 86 :00
68A0  04 0D 1E 26 0E 0D 21 27 :B8
68A8  08 9E 12 10 9E 06 10 AF :2B
68B0  83 96 17 8D 04 8D 17 20 :85
68B8  AF 0D 26 26 04 0F 06 97 :B8
68C0  07 96 0B 9B 07 97 0B 96 :82
68C8  0A 99 06 97 0A 39 AE 9F :D0
68D0  6F 0E 9F 0E 39 D6 17 C1 :11
68D8  02 27 02 C6 04 0F 06 0F :19
68E0  07 9D 80 81 24 27 31 BD :DE
68E8  60 3A 25 03 39 9D 80 BD :D5
68F0  60 61 24 03 1A 01 39 BD :F9
68F8  60 3A 24 40 34 04 1F 89 :DE
6900  4F 34 06 DC 06 58 49 58 :64
6908  49 58 49 D3 06 D3 06 E3 :7F
6910  E1 DD 06 35 04 20 D6 39 :2C
6918  9D 80 BD 60 61 24 01 39 :F9
6920  BD 60 32 24 17 48 48 48 :62
6928  48 34 04 C6 04 48 09 07 :A2
6930  09 06 5A 26 F8 35 04 5A :1A
6938  2A DE 1C FE 39 8E 70 00 :59
6940  9F 0A 9E 44 9F 0E 9E 0A :E0
6948  9F 28 AE 9F 6F 0E 10 27 :C8
6950  03 EE 9E 0E 1F 13 C6 06 :9B
6958  9D 80 81 20 27 09 25 03 :16
6960  5A 2A F5 0C 1E 20 03 BD :83
6968  61 F8 0D 1E 27 07 86 04 :3C
6970  97 17 16 02 AC 0D 21 10 :B0
6978  26 02 CE 0D 23 10 27 00 :5D
6980  ED 96 18 84 FE 81 1E 27 :E3
6988  79 84 FC 81 34 27 28 9D :9A
6990  80 81 27 26 14 9D 80 4D :CC
6998  27 18 97 07 9D 80 81 27 :A2
69A0  26 10 9D 80 BD 60 61 20 :F1
69A8  07 33 5F C6 02 BD 68 DD :63
69B0  25 02 0C 1E 16 02 27 0F :9F
69B8  07 8D 33 D6 06 D7 18 24 :B6
69C0  28 C4 FE 81 08 26 06 C1 :60
69C8  34 27 1E 20 08 81 09 26 :51
69D0  04 C1 36 27 14 8E 6A 5A :88
69D8  8D 61 9A 07 97 07 9D 80 :4A
69E0  81 2C 27 D5 BD 60 61 25 :4C
69E8  02 0C 1E 16 01 F0 96 18 :E1
69F0  97 06 BE 6A 5A 9F 00 8E :4C
69F8  6A 46 86 02 BD 60 6E 1F :E2
6A00  13 39 8D EA 24 E3 8D 30 :87
6A08  48 48 48 48 97 07 9D 80 :DB
6A10  81 2C 26 D5 8D DC D6 06 :ED
6A18  D7 18 24 CD 8D 1A 9A 07 :28
6A20  97 07 2B 06 84 08 26 C1 :42
6A28  20 04 84 08 27 BB 9D 80 :AF
6A30  BD 60 61 24 B4 16 01 A6 :13
6A38  8E 6A 64 4D 27 05 30 01 :06
6A40  4A 26 FB A6 84 39 43 43 :54
6A48  41 00 42 00 44 50 58 00 :6F
6A50  59 00 50 43 44 00 35 00 :65
6A58  55 00 01 02 04 08 10 20 :94
6A60  80 06 40 40 0A 08 09 0B :2C
6A68  01 02 05 00 04 03 0D 20 :3C
6A70  10 27 01 AD 9D 80 81 2A :AD
6A78  26 2E 9D 80 81 20 26 11 :49
6A80  0D 1D 27 0D 9D 80 81 2D :29
6A88  27 0A 33 5F BD 68 D5 25 :E2
6A90  0A 16 00 B6 BD 68 D5 24 :F4
6A98  F8 8D 02 20 64 03 06 03 :17
6AA0  07 9E 06 30 01 9F 06 39 :BA
```

```
6AA8  33 5F 81 24 27 05 BD 60 :80
6AB0  3A 24 50 0D 22 27 0A C6 :D4
6AB8  02 BD 68 DD 25 3F 16 00 :7E
6AC0  89 BD 68 DB 25 1A 16 00 :DE
6AC8  B2 96 07 90 17 97 07 24 :B8
6AD0  02 0A 06 96 07 90 0B 97 :E1
6AD8  07 96 06 92 0A 97 06 39 :15
6AE0  0D 1D 27 19 8D E3 96 17 :87
6AE8  81 02 26 11 96 06 27 09 :86
6AF0  43 26 57 0D 07 2B 06 20 :25
6AF8  51 0D 07 2B 4D 27 02 8D :93
6B00  D2 20 49 0F 16 9E 10 9F :AD
6B08  12 9C 14 26 0A 0D 16 27 :3C
6B10  39 9E 04 1F 13 20 58 C6 :4B
6B18  06 1F 31 9F 02 9E 02 A6 :3D
6B20  06 BD 60 51 25 17 81 20 :51
6B28  25 20 30 01 9F 02 9E 12 :C7
6B30  A1 84 26 1A 30 01 9F 12 :47
6B38  5A 2A E2 20 0D 9E 12 5D :A0
6B40  27 04 6D 84 26 08 0D 16 :6D
6B48  27 0D 0C 1E 20 3C CB 02 :87
6B50  30 01 5A 26 FB 20 B0 0C :88
6B58  16 5D 27 05 30 01 5A 26 :50
6B60  FB 9F 12 AE 84 9F 06 9E :21
6B68  02 9F 04 9E 12 20 DF 9D :F1
6B70  80 BD 60 61 25 46 BD 60 :86
6B78  59 25 2A 81 5D 26 0D 0D :C6
6B80  1F 27 C7 9D 80 BD 60 61 :A8
6B88  24 C0 20 52 81 2C 26 BA :E3
6B90  0D 25 27 B6 BD 6A C9 96 :95
6B98  17 D6 18 C1 20 24 01 4A :55
6BA0  81 03 16 FF 45 9E 06 9F :21
6BA8  04 81 2D 27 12 BD 68 DB :EB
6BB0  25 08 34 02 8D 04 35 02 :2B
6BB8  20 C1 8D 14 16 FF 21 BD :75
6BC0  68 DB 25 08 34 06 8D 0F :46
6BC8  35 06 20 AF 8D 09 20 EC :AC
6BD0  DC 06 D3 04 DD 06 39 DC :B1
6BD8  04 D3 06 DD 06 39 96 17 :A6
6BE0  8E 6F 18 0D 23 26 10 0D :88
6BE8  22 26 0C 0D 25 26 14 0D :CD
6BF0  1D 27 1F 81 02 26 1B 4A :71
6BF8  30 01 4A 26 FB 96 07 A7 :E0
6C00  84 20 1E 34 02 D6 18 C1 :A7
6C08  20 24 01 4A 81 03 35 02 :4A
6C10  27 E5 80 02 27 05 30 01 :EB
6C18  4A 26 FB 10 9E 06 10 AF :DE
6C20  84 8E 6F 18 9F 00 BD 68 :5D
6C28  14 96 17 BD 68 B9 0D 24 :D0
6C30  26 17 0D 27 26 13 96 17 :57
6C38  27 0F 9E 00 E6 80 9F 00 :D9
6C40  9E 04 E7 80 9F 04 4A 26 :1C
6C48  F1 0D 2C 26 12 0D 1E 26 :B3
6C50  06 BD 68 CE 16 FC EF 8D :87
6C58  1C 8D 4E 26 FC 20 F2 8E :B9
6C60  6F 28 BD 6E 48 0D 1E 27 :5C
6C68  49 C6 03 86 2A BD 6E 12 :FF
6C70  5A 26 FA 20 68 9E 0E 30 :DE
6C78  02 34 36 10 AE 84 C6 04 :78
6C80  8E 6C A1 4F 1E 02 A3 84 :31
6C88  2B 05 1E 02 4C 20 F5 E3 :94
6C90  81 1E 02 8B 30 17 01 7A :EE
6C98  5A 26 E8 35 36 BD 6E 75 :73
6CA0  39 03 E8 00 64 00 0A 00 :92
6CA8  01 A6 04 30 01 BD 6E 12 :19
6CB0  4D 39 9E 28 9F 04 0F 06 :04
6CB8  D6 17 27 21 BD 6E 4F 0C :BB
6CC0  06 96 06 81 05 26 13 5A :BB
6CC8  27 13 5C 0F 06 86 0D BD :FB
6CD0  6E 12 34 04 C6 05 8D 5E :6E
```

```
6CD8  35 04 5A 26 DF C6 14 8D :FF
6CE0  55 8D 92 10 9E 0E A6 26 :FC
6CE8  81 2A 27 43 8D BB 27 30 :B4
6CF0  81 20 26 F8 C6 20 8D 3E :70
6CF8  A6 03 81 27 26 0A 8D A9 :B7
6D00  27 1E A6 03 81 27 26 F6 :B2
6D08  A6 04 81 20 27 15 D6 40 :9D
6D10  C1 4A 26 07 BD 6E 75 C6 :9E
6D18  2E 8D 1B BD 6C A9 26 D8 :A6
6D20  16 FF 2E A6 03 81 20 26 :B3
6D28  E5 C6 2C 8D 09 20 DF BD :29
6D30  6C A9 26 FB 20 EA D1 40 :51
6D38  2E 01 39 BD 6E 75 20 F6 :1E
6D40  0D 2C 27 02 8D 02 35 FF :25
6D48  BD 6E 7A 8E 6F 36 6F 80 :C7
6D50  8C 6F 3C 26 F9 BD 6E 7A :FB
6D58  C6 05 8E 6F 30 A6 06 A7 :4B
6D60  80 8C 6F 36 26 F7 86 FF :53
6D68  A7 80 8C 6F 3C 26 F9 9E :1B
6D70  10 9F 12 9C 14 26 31 86 :4E
6D78  FF 91 36 26 01 39 8E 6F :23
6D80  06 BD 6E 48 34 04 8E 6F :AE
6D88  36 C6 06 A6 84 26 02 86 :DA
6D90  20 BD 6E 12 30 01 5A 26 :0E
6D98  F2 C6 03 BD 6E 75 5A 26 :DB
6DA0  FA 35 04 5A 26 B4 20 AD :34
6DA8  9F 02 8E 6F 36 9F 04 1C :93
6DB0  FE 9E 04 8C 6F 30 27 0E :00
6DB8  A6 82 9F 04 9E 02 A2 05 :12
6DC0  30 1F 9F 02 20 EB 24 3E :5D
6DC8  9E 12 9F 02 8E 6F 3C 1C :A6
6DD0  FE 9F 04 9E 02 A6 05 30 :1C
6DD8  1F 9F 02 9E 04 A2 82 9F :25
6DE0  04 8C 6F 36 26 EB 24 1E :88
6DE8  9E 12 9F 02 8E 6F 36 9F :23
6DF0  04 9E 02 A6 80 9F 02 9E :09
6DF8  04 A7 80 8C 6F 3C 26 EF :77
6E00  9E 12 AE 06 9F 06 86 08 :97
6E08  9E 12 30 01 4A 26 FB 7E :CA
6E10  6D 71 34 16 4D 27 06 81 :23
6E18  0D 27 02 20 0B 8E 6E 36 :93
6E20  0F 40 AD 9F FB FA 20 0C :BC
6E28  B7 6E 44 8E 6E 3E AD 9F :EF
6E30  FB FA 0C 40 35 96 14 00 :20
6E38  6E 3C 00 02 0A 0D 14 00 :D7
6E40  6E 44 00 01 00 00 35 94 :7C
6E48  8D 0A 8D 08 8D 27 39 8D :A6
6E50  03 8D 22 39 34 04 A6 80 :49
6E58  1F 89 84 F0 C4 0F 44 44 :77
6E60  44 44 8D 06 1F 98 8D 02 :61
6E68  35 84 81 0A 2B 02 8B 07 :03
6E70  8B 30 16 FF 9D 86 20 16 :29
6E78  FF 98 86 0D 16 FF 93 A6 :78
6E80  84 81 04 26 01 39 17 FF :7F
6E88  89 30 01 20 F2 39 00 00 :05
6E90  00 00 00 00 00 00 00 00 :00
6E98  00 00 00 00 00 00 00 00 :00
6EA0  00 00 00 00 00 00 00 00 :00
6EA8  00 00 00 00 00 00 00 00 :00
6EB0  00 00 00 00 00 00 00 00 :00
6EB8  00 00 00 00 00 00 00 00 :00
6EC0  00 00 00 00 00 00 00 00 :00
```

超初心者のための
マシン語の入力の仕方

① MONコマンドでモニタコマンドレベルにし，Mコマンドで書き込みます。たとえば38ページのリスト1なら

MON⏎
＊D3000⏎
3000　00—34⏎
3001　00—76⏎
3002　00—AE⏎

と，以下$3031番地まで入力します。

② 本誌では，長いプログラムについてはチェックサム付きダンプリストで掲載しています。第3号の「Oh!FM用チェックサムプログラム」をRUNさせて入力すれば，1行分入力するごとに，チェックサムを確認できます。入力ミス防止のため，ぜひご使用ください。

なお，岡崎市の小笠原克彦氏より，A～Fをテンキー部で入力するプログラムが送られてきましたので，変更点のみリストAに示します（FM-7,11用)。

＊　＊

リストA

```
625 IF I$="*" OR I$="/" OR I$="+" OR I$="-" OR I$="=" OR I$="," THEN GOSUB 900
900 '****** hex key
910 IF I$=CHR$(42) THEN I$="A" : RETURN
920 IF I$=CHR$(47) THEN I$="B" : RETURN
930 IF I$=CHR$(43) THEN I$="C" : RETURN
940 IF I$=CHR$(45) THEN I$="D" : RETURN
950 IF I$=CHR$(61) THEN I$="E" : RETURN
960 IF I$=CHR$(44) THEN I$="F" : RETURN
```

マシン語特集partⅡ

マシン語プログラム実例集1 4

ソート＆画面操作

鶴岡哲明

前号では算術演算をマシン語で実現した例をいくつか挙げたが，今回は引き続きソート・画面操作などの例を挙げたいと思う。

例としては平凡だが，マシン語プログラムをどう組むかのコツみたいなものはわかっていただけると思う。またZ80などほかのCPUのマシン語をやっている人には，6809のインデックスの強力さもわかっていただけると思う。

また，前回に引き続きアセンブリ言語の説明も必要に応じて触れたいと思う。

ソートプログラム

6809CPUはインデックスアドレッシングが柔軟にできるため，こういう大量のデータを操作するのが容易である。そのため，BASICと比べてかなり最適化を行うことができ，処理速度が格段に違う。

今回取り上げたのは単純選択法で，最も小さいものを探し，それを先頭に入れ，2番目以降についても同じことをし，……というようにして並びかえる方法で，最もわかりやすい方法である。ただし，いちばん小さいものを選んだときに前のデータを消しては困るので，交換するという形をとって残すようにする。

これをマシン語にしたものが**リスト1**である。また同様のことをBASICでも行い，時間を比べることにした。まず，前もって1000個の整数(−32768〜32767)を乱数で作り，DATAというファイルを作った。そのあと**リスト2**のプログラムを走らせたところ，なんとBASICでは4485秒もかかったが，マシン語でやったら15秒で終えてしまった(FM-8の場合。FM-7ではマシン語のみ試したが9秒で済んだ。)

さて，**リスト1**のプログラムの仕様であるが，できるだけ汎用のものをと思い，X(0)にデータ数，X(1)以後にデータを入れて，USR(VARPTR(X(0)))を実行すればよい。このソート法はデータ数の2乗に比例する時間がかかるので，100個程度のソートはあっという間であろう。

では解説に移ろう。

リスト1のプログラムはデータトップアドレス−2の値(つまりX(0)のアドレス)にX(0)の値の2倍を加え，これをデータエンドアドレスとしてスタックに入れる。また，データエンドアドレス＋2も終了条件を判断するのに必要なので，これも前もって計算しスタックに入れる。あとはXレジスタの示すアドレスが以降のデータの中で最も小さくなるよう比較・交換を行う。

この比較でも6809の威力が発揮される。符号付き比較が楽なうえ，これが16ビットでも適用できるのである。そのためZ80で組むよりずっと楽である。

詳しくはBASICと対比させればわかるであろう。メモリ↔レジスタの交換はできないので，Uレジスタをやむなく使っている。また，X＋2の値をYレジスタに入れるにはLEA命令を使う。2倍の計算はシフトで行っている。

さて，これは整数型であるが，浮動小数点の場合もそう複雑にならないと思う。少し比較に手間がかかる程度だろう。

マシン語による画面の操作

一口にマシン語で画面を操作するといっても，VRAMがサブCPU管理のため，BIOSのお世話になるかしてサブCPUに指令を送ることしかメイン側からはできない。しかし，サブCPU側に送る指令の中に「このアドレスのサブルーチンに飛べ」というものがあり，これを用いれば自分のプログラムを実行させることができる。今回はサブCPUがROM内に持っているLINEを用いる方法と，直接サブCPUにプログラムを実行させる方法の2例を取り上げてみた。

(1) LINE

LINEはBIOSをコールすれば容易である。BIOSの使い方は他にもいろいろ解説されているので，ここでは最小限の解説にとどめさせていただく。BIOSコールでも，サブCPUに指令するデータを送るときの方法を**図1**に示す。このようなデータブロックの先頭アドレスにRCBという名を付け，この値をXレジスタに入れてBIOSを呼ぶ。また，サブCPUに送られるコマンドはサブコマンドと呼ばれさまざまなものがあるが，今回は＄15(LINE)である。これと必要なパラメータを**図2**のようにセットすればよい。あとはBIOSが共有RAMにデータを書いてサブCPUを走らせてくれる。

リスト1

○ここで扱われるプログラムは，個人で利用するほかは著作権法上，無断複製を禁じられています。
COPY RIGHT © 1983 TETSUMEI TSURUOKA

```
PAGE  001 (       ,000436) SORT

00100                             NAM    SORT
00110                      ********************************
00120                      *      SORT PROGRAM v0.3        *
00130                      ********************************
00140                             OPT    NOM
00150  3000                       ORG    $3000
00160  3000 34   76               PSHS   X,D,U,Y
00170  3002 AE   02               LDX    2,X       ;load DATA TOP ADDRESS-2
00180  3004 34   10               PSHS   X
00190  3006 EC   81               LDD    ,X++
00200  3008 58                    ASLB
00210  3009 49                    ROLA
00220  300A E3   E1               ADDD   ,S++      ;DATA END ->D
00230  300C 34   06               PSHS   D
00240  300E C3   0002             ADDD   #2        ;DATA END+2 ->D
00250  3011 34   06               PSHS   D
00260  3013 EC   84        XLOOP  LDD    ,X
00270  3015 31   02               LEAY   2,X
00280  3017 10A3 A1        YLOOP  CMPD   ,Y++
00290  301A 2F   06   3022        BLE    NSWAP
00300                      *
00310                      *  SWAP D,(Y)
00320                      *
00330  301C EE   3E               LDU    -2,Y
00340  301E 1E   03               EXG    D,U
00350  3020 EF   3E               STU    -2,Y
00360  3022 10AC E4        NSWAP  CMPY   ,S
00370  3025 26   F0   3017        BNE    YLOOP
00380  3027 ED   81               STD    ,X++
00390  3029 AC   62               CMPX   2,S
00400  302B 26   E6   3013        BNE    XLOOP
00410  302D 32   64               LEAS   4,S
00420  302F 35   76               PULS   X,D,Y,U
00430  3031 39                    RTS
00440                             END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=3000
PROGRAM END   ADDR=3031
PROGRAM ENTRY ADDR=****
```

リスト2

```
10 CLEAR300,&H2FFF:LOADM"SORT3"
20 DEFUSR=&H3000
30 DEFINT A-Z
40 DIM X(1000):OPEN"I",1,"DATA":X(0)=1000
50 FOR I=1 TO 1000
60  INPUT #1,X(I)
70 NEXT:TIME$="00:00:00"
80 A=USR(VARPTR(X(0)))
90 PRINTTIME:CLOSE1
100 OPEN"I",1,"DATA"
110 FOR I=1 TO 1000
120  INPUT #1,X(I)
130 NEXT
140 TIME$="00:00:00"
150 FOR I=1 TO 999
160  FOR J=I+1 TO 1000
170   IF X(I)>X(J) THEN SWAP X(I),X(J)
180  NEXT
190 NEXT
200 PRINTTIME
210 'M:0015
220 'B:4485
```

さて，プログラム例を**リスト3**に示す。これは**リスト4**をマシン語化したものだ。

リスト3がどういう内容かは一目でわかるだろう。カラーコードの計算はMODをとるのが面倒なので，Bレジスタを使って7の次は1になるよう判断し，1回線を引くごとにBレジスタを1ずつ増やしている。なお，**リスト3**のプログラムはリロケータブルになるように細工してある。Yレジスタにラベル TDAのアドレスをロードして，データブロックはすべてインデックスで指定している。また，この値をYレジスタにロードするのはPCリラティブを用いている。

行番号290のRTSまでがプログラムで，あとはデータエリアとなっているが，一部定数となるのであらかじめ設定しておく必要がある。打ち込むときは最後まで入れて

リスト3

```
PAGE  001 (       ,000125) LINE

00010                            NAM    LINE
00020                     * ver 0.3
00030                            OPT    0
00040  3000                      ORG    $3000
00050            FBFA       BIOS  EQU    $FBFA
00060  3000 31   8C 41           LEAY   <TDA,PCR
00070  3003 10AF 8C 4D           STY    <TOP,PCR
00080  3007 CC   00C7            LDD    #199
00090  300A ED   A8 0B           STD    <Y2-TDA,Y
00100  300D CC   0000            LDD    #0
00110  3010 E7   A8 03           STB    <CCODE-TDA,Y
00120  3013 ED   A8 07           STD    <Y1-TDA,Y
00130  3016 ED   A8 05      LOOP  STD    <X1-TDA,Y
00140  3019 CC   027F            LDD    #639
00150  301C A3   A8 05           SUBD   <X1-TDA,Y
00160  301F ED   A9 0009         STD    X2-TDA,Y
00170  3023 E6   A8 03           LDB    <CCODE-TDA,Y
00180  3026 5C                   INCB
00190  3027 C1   08              CMPB   #8
00200  3029 25   02   302D       BLO    PASS
00210  302B C6   01              LDB    #1
00220  302D E7   A8 03      PASS  STB    <CCODE-TDA,Y
00230  3030 30   8C 1F           LEAX   <RCB,PCR
00240  3033 AD   9F FBFA         JSR    [BIOS]
00250  3037 EC   A8 05           LDD    <X1-TDA,Y
00260  303A C3   0001            ADDD   #1
00270  303D 1083 0280            CMPD   #640
00280  3041 25   D3   3016       BLO    LOOP
00290  3043 39                   RTS
00300  3044      0002       TDA   RMB    2
00310  3046      15              FCB    $15      ;code LINE
00320  3047      0001       CCODE RMB    1
00330  3048      00         FCODE FCB    0
00340  3049      0002       X1    RMB    2
00350  304B      0002       Y1    RMB    2
00360  304D      0002       X2    RMB    2
00370  304F      0002       Y2    RMB    2
00380  3051      00         OPT   FCB    0
00390  3052      10         RCB   FCB    $10
00400  3053      0001       STS   RMB    1
00410  3054      0002       TOP   RMB    2
00420  3056      000E       LDA   FDB    RCB-TDA
00430                            END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=3000
PROGRAM END   ADDR=3057
PROGRAM ENTRY ADDR=****
```

ほしい。実行はEXEC&H3000である。

ここで新しい疑似命令が出たので紹介しておこう。

FCB……定数定義の疑似命令。オペランド１バイトを定義する。たとえばリスト３の行番号 310は＄3046番地に＄15を入れることを示す。

FDB……2バイトの定数定義の疑似命令。内容はFCBと同じである。

また，アセンブラはラベルの代わりに計算式も使える。たとえば**リスト３**の行番号90では，

STD　<Y2−TDA, Y

となっているが，これは Y2 −TDA＝7となり，

STD　7, Y

の意味となる。なお，使える計算は＋，−，＊，÷程度である。これもアセンブラによって多少異なる。

数値の前の<は８ビットという意味で，付けないと８ビットで収まっても16ビットで表現される場合がある。強制的に16ビットにしたいときは>を使う。

さて，実行時間であるが，サブCPUの仕事がほとんどなので，実行時間はほぼ同じで，160秒対162秒(BASICの方が遅い。FM-8の場合。FM-7はちょうどこの半分の時間であった)である。

リスト４

```
10 FOR I=0 TO 639
20   LINE(I,0)-(639-I,199),PSET,I MOD 7+1
30 NEXT
```

(2)サブCPUへの命令転送

サブCPUに送るサブコマンドでできることは限られており，しかもBASICで画面関係のステートメントを使うのと速度的にも差がない。そこで，直接サブCPUにプログラムを実行させたくなる。実はそのためのサブコマンドが用意されていて，これはサブCPUをきめ細かく使うには欠かせない。前号のマシン語特集でも触れられているので，ここでは簡単に触れておく。

これも共有RAMに送られるデータを作り，BIOSコールをするのはLINEの場合と同じである。**図3**を見てほしい。プログラムの例を**リスト5，6**に示す。**リスト5**は面白くも何ともないが，メインテナンスコマンドの使い方がよくわかると思う。行番号110～160はXレジスタをセットしてBIOSコールをする。行番号200からは共有RAMに送るデータである。なお，FCCはアスキーコードを入れる疑似命令で，文字列の初めと終りを同じ文字で囲んで使用する。この例では“を用いたが，／の方がよく用いられる。

行番号220ではサブルーチンコールのアドレスを設定している。このアドレスは共有RAM上のアドレス（サブCPUから見て）となっている。つまり，転送データの中にプログラムを入れてしまうわけだ。ラベルSUBRの命令がちょうどこのアドレスになる。この例では本来$D38F～のプログラムが$3017～に入れてあるので，OFFSETというラベルに，このずれ分を入れる。

見ればおわかりいただけると思うが，これはVRAMの1と0を反転させるプログラムである。FM-7，8のVRAMは**図4**のよううになっていて，各ビットを反転させると，だんだん色がかわっていくのがわかると思う。

そして**リスト6**は，青の部分を指定したパターンに書き換えるものである。これを用いれば，さまざまな中間色を表示できる。

使い方はタイルパターン（6バイト）を文字列に入れ(青，赤，緑の順)，USR関数で呼び出すと青を指定したパターンに書き換える。パターン指定法はPC-8801などと同様である。

では処理内容の説明に入ろう。まずUレジスタであるが，これは読み出すタイルパターン用のポインタである。またBレジス

図1　BIOSコール(サブCPUへコマンドを送る)

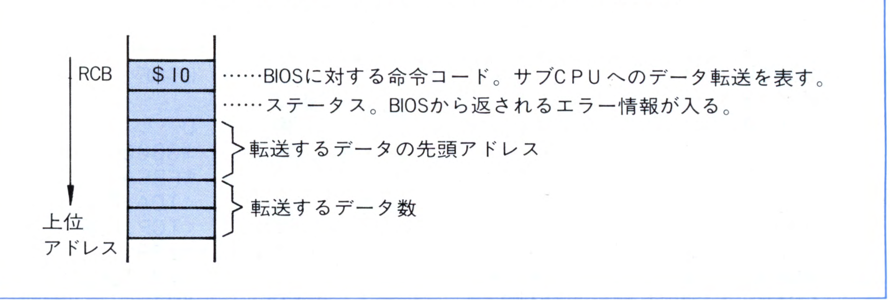


図2　LINEの転送データ

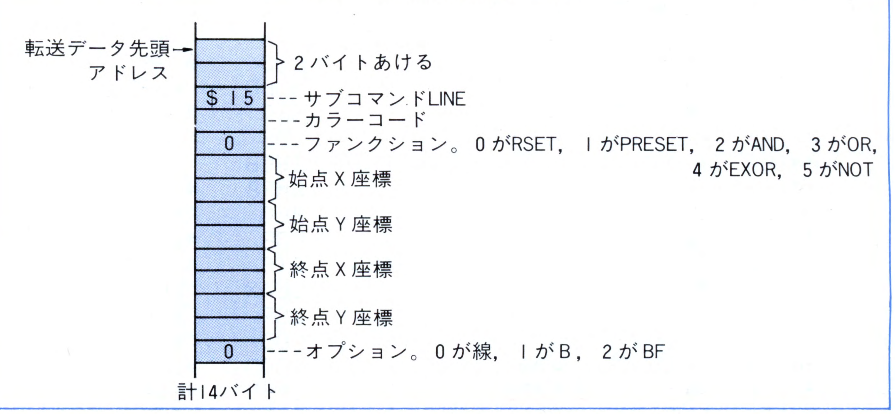
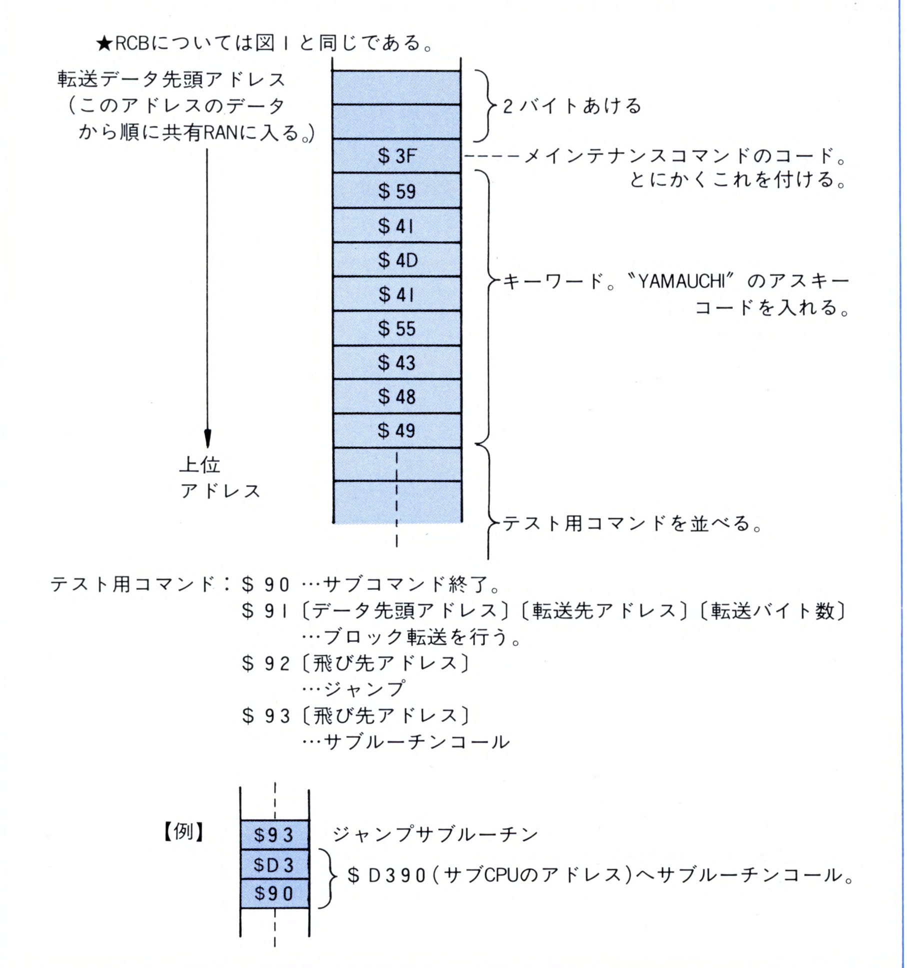


図3　サブCPUに直接プログラムを実行させる方法



リスト 5

```
PAGE  001 (       ,002351) SCALL2

                                NAM    SCALL2
                                OPT    0
3000                            ORG    $3000
            FBFA      BIOS      EQU    $FBFA
3000 8E     3028                LDX    #RCB
3003 AD     9F FBFA             JSR    [BIOS]
3007 39                         RTS
3008        0002      TDA       RMB    2
            A378      OFFSET    EQU    $D380-TDA
300A        3F                  FCB    $3F      ;code YAMAUCHI
300B        59                  FCC    'YAMAUCHI'
300C        41
300D        4D
300E        41
300F        55
3010        43
3011        48
3012        49
3013        93                  FCB    $93
3014        D38F                FDB    SUBR+OFFSET
3016        90                  FCB    $90
3017 B6     D409      SUBR      LDA    $D409
301A 8E     0000                LDX    #$0000
301D 63     80        LP        COM    ,X+
301F 8C     C000                CMPX   #$C000
3022 26     F9   301D           BNE    LP
3024 B7     D409                STA    $D409
3027 39                         RTS
3028        10        RCB       FCB    $10
3029        0001      STS       RMB    1
302A        3008                FDB    TDA
302C        0020      LDA       FDB    RCB-TDA
                                END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=3000
PROGRAM END   ADDR=302D
PROGRAM ENTRY ADDR=****
```

リスト 6

```
                                NAM    SCALL3-R
                      ***********************************
                      *  C-COLOR WITH HARF COLOR        *
                      *    copyright (c) by T.Tsuruoka  *
                      ***********************************
                                OPT    0
3000                            ORG    $3000
            FBFA      BIOS      EQU    $FBFA
3000 34     36                  PSHS   D,X,Y
3002 C6     06                  LDB    #6
3004 AE     01                  LDX    1,X
3006 31     8D 0094             LEAY   DTBL0,PCR
300A A6     80        MOR       LDA    ,X+
300C A7     A0                  STA    ,Y+
300E 5A                         DECB
300F 26     F9   300A           BNE    MOR
3011 30     8D 000F             LEAX   TDA,PCR
3015 AF     8D 008D             STX    SAD,PCR
3019 30     8D 0087             LEAX   RCB,PCR
301D AD     9F FBFA             JSR    [BIOS]
3021 35     36                  PULS   D,X,Y
3023 39                         RTS
3024        0002      TDA       RMB    2
            A35C      OFFSET    EQU    $D380-TDA
3026        3F                  FCB    $3F      ;code YAMAUCHI
```

```
3027         59                  FCC    'YAMAUCHI'
3028         41
3029         4D
302A         41
302B         55
302C         43
302D         48
302E         49
302F         93                  FCB    $93
3030         D38F                FDB    SUBR+OFFSET
3032         90                  FCB    $90
3033 B6      D409         SUBR   LDA    $D409
3036 8E      0000                LDX    #$0000
3039 1F      13                  TFR    X,U
303B C6      50           LP1    LDB    #80
303D 31      89 8000      LP2    LEAY   $8000,X
3041 A6      A4                  LDA    ,Y
3043 AA      89 4000             ORA    $4000,X
3047 43                          COMA
3048 A4      80                  ANDA   ,X+
304A 27      39    3085          BEQ    PASS
304C 34      02                  PSHS   A         ;*1PAINT
304E 43                          COMA
304F 34      02                  PSHS   A         ;*2 NPAINT
3051 A4      A4                  ANDA   ,Y
3053 34      02                  PSHS   A         ;*3RPAINT
3055 A6      62                  LDA    2,S
3057 A4      C9 D3FC             ANDA   DTBL+2,U  ;RPDATA
305B AA      E0                  ORA    ,S+       ;*2
305D A7      A4                  STA    ,Y
305F A6      E4                  LDA    ,S
3061 A4      89 3FFF             ANDA   $3FFF,X
3065 34      02                  PSHS   A         ;*3
3067 A6      62                  LDA    2,S
3069 A4      C9 D3FB             ANDA   DTBL+1,U
306D AA      E0                  ORA    ,S+       ;*2
306F A7      89 3FFF             STA    $3FFF,X
3073 A6      E4                  LDA    ,S
3075 A4      1F                  ANDA   -1,X
3077 34      02                  PSHS   A'        ;*3
3079 A6      62                  LDA    2,S
307B A4      C9 D3FA             ANDA   DTBL,U
307F AA      E4                  ORA    ,S
3081 A7      1F                  STA    -1,X
3083 32      63                  LEAS   3,S
3085 5A                   PASS   DECB
3086 26      B5    303D          BNE    LP2
                          *    1 LINE END
3088 33      43                  LEAU   3,U
308A 11B3    D3F8                CMPU   NDATA
308E 26      03    3093          BNE    NCU
3090 CE      0000                LDU    #0
3093 8C      3E80         NCU    CMPX   #80*200
3096 26      A3    303B          BNE    LP1
3098 B7      D409                STA    $D409
309B 39                          RTS
309C         0006         NDATA0 FDB    6
309E         55           DTBL0  FCB    $55
             D3F8         NDATA  EQU    NDATA0+OFFSET
             D3FA         DTBL   EQU    DTBL0+OFFSET
309F         33                  FCB    $33
30A0         FF                  FCB    $FF
30A1         AA                  FCB    $AA
30A2         CC                  FCB    $CC
30A3         FF                  FCB    $FF
30A4         10           RCB    FCB    $10
30A5         0001         STS    RMB    1
30A6         3024         SAD    FDB    TDA
30A8         0080         LDA    FDB    RCB-TDA
                                 END
```

```
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=3000
PROGRAM END   ADDR=30A9
PROGRAM ENTRY ADDR=****
```

図4

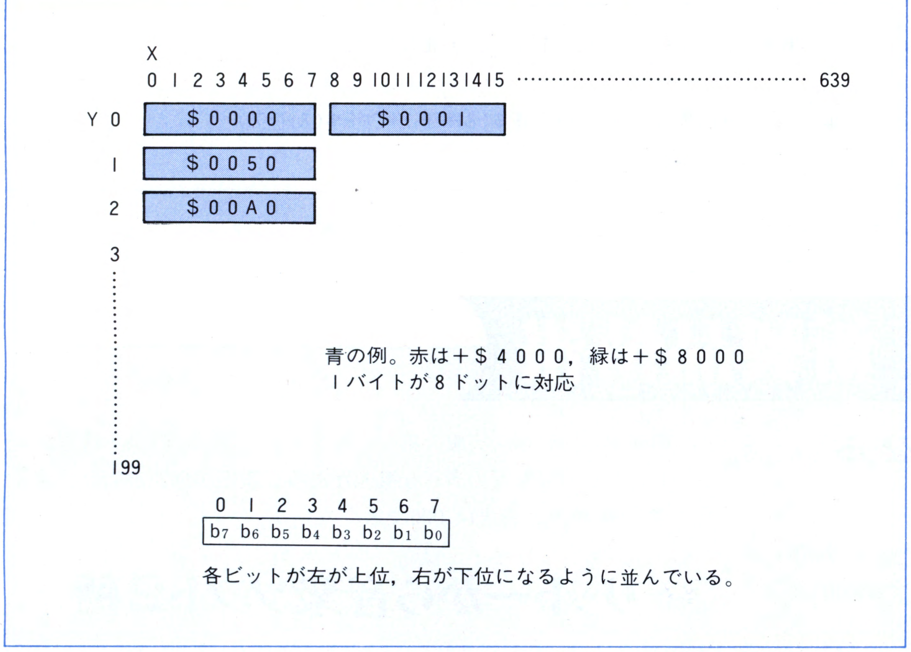
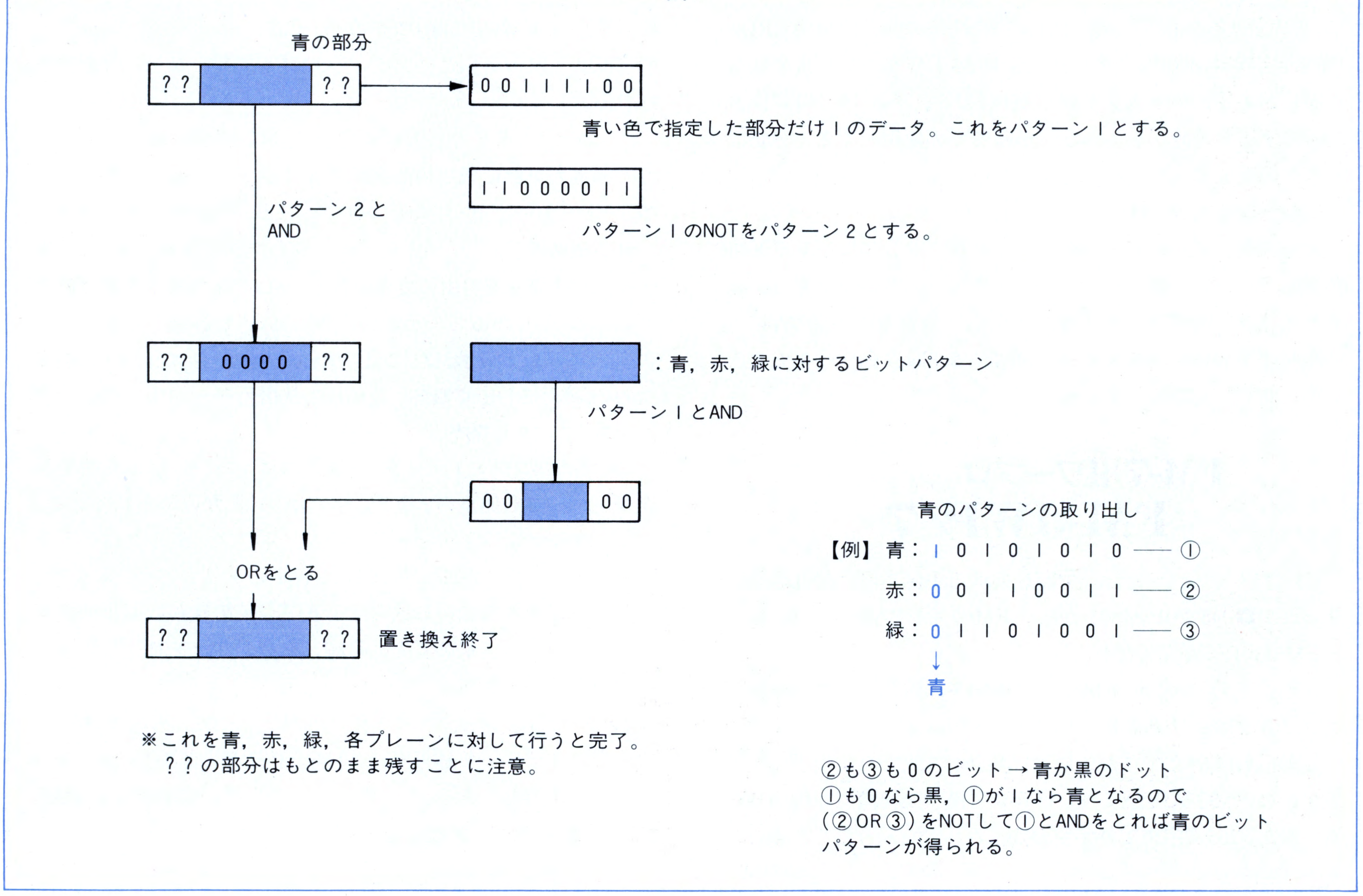


図5



タは80バイト(横の640ドット)分のカウンタで，Bが0になると，読み出すタイルパターンを変える。

VRAMで青である部分の判定は，青のビットが1，他が0の点を見つければよい。これを指定したパターンで置き換える。この置き換えがなかなか手間がかかる。これは**図5**のようにすればよい。

以上で色変更完了である。使用例として**リスト7**を挙げておく。なお**リスト6**はリロケータブルになっているので，BASICテキストエリアの邪魔にならないところに置くとよいだろう。

*　*　*

さて，今回の基本テクニック集，いかがであろうか。できる限りリロケータブルにしたので実用にもなると思う。ワークエリアもほとんどスタックを使用しているので，

リスト7

```
10 LOADM"SCALL3"
20 DEFUSR=&H3000:A%=TIME-INT(TIME/65536!)*65536!-32768!:RANDOMIZE(A%)
30 DEFFNR(X)=INT(RND(1)*X)
40 FOR I=1 TO 3:FOR J=1 TO 3
50 X=I*160:Y=J*50
60 CIRCLE (X,Y),150,7,,,,N
70 NEXT:NEXT
80 FOR I=0 TO 30:X=FNR(640):Y=FNR(200):C=FNR(10)-2:IFC<2 THEN C=1
85 PAINT(X,Y),C,7
90 NEXT
100 A$=CHR$(&H22)+CHR$(&HFF)+CHR$(&HEE)+CHR$(&H11)+CHR$(&HFF)+CHR$(&H77)
110 A$=USR(A$)
```

ワークエリアが重なってだめ，ということはないと思う。また，こうしたプログラムにするにはどうしたらよいかなども，わかってもらえたのではないかと思う。

（参考文献　インターフェース'82.6月号）

新製品情報

シングルフロッピディスクセット FD-7

FM-7用のシングルミニディスクFD-7（定価99,900円）が，マイクロシステムズ(株)(〒101 東京都千代田区外神田1-7-6三神ビル☎03-251-5148）よりこのほど発売される。

FM-7内にフロッピディスク・インタフェース（MACRO-7-5)を実装するタイプで,同時発売の増設用ミニディスクMACRO-FD5（定価74,800円)との併用により，2ドライブに拡張できる。また，8インチディスク（MACRO-FD8，予価149,800円）との同時使用も可能。このほか，単密度での使用もでき，純正ディスクの上位コンパチ機。

主な仕様は次のとおり。

記録方式：MFM，FM／最大記憶容量：アンフォーマット時500KB／トラック数：80（40×2）トラック／トラック密度：48TPI／トラック間アクセス時間：5ms／消費電力：20VA以下／外形寸法：150×300×45mm／重量：3.5kg／動作時推薦温度：10～35℃／同湿度：20～80%

FM-7用ワープロ FM-JWP/7

(株)エス・ビー・シー（〒231 横浜市中区本町5-49丸忠センタービル☎045-201-8480）から，FM-7用日本語ワープロソフトFM-JWP/7が発売された。

5インチディスケット版で，定価20,000円と低価格。使用可能プリンタはシリアルドットプリンタのみ。

最大桁数は59文字だが，最大行数は240行。カナ漢字変換のほか，ローマ字入力ができ，ローマ字漢字変換機能が付いている。また，255文字まで外字登録ができ，文節入力もできる。このほか，けい線（実線・細線・点線など），計算，移動・複写，センタリングなどの多彩な機能がある。漢字・熟語辞書は，漢字6,000読み，熟語14,000読み。

リットーから音楽ソフト2種

(株)リットーミュージック（〒160 東京都新宿区四谷1－5新四谷駅前ビル☎03-359-0266)から，FM-7用の「ミュージック・ハーモナイザー3」と「デジ・ソング・エディター」が新発売される。

「ミュージック・ハーモナイザー3」は，作曲・編曲・プログラム演奏を気軽に楽しめる音楽創作ソフト。メロディを入力するとFM-7がコード・伴奏をつけてくれる。伴奏のリズムパターンは12種類あり，オリジナルパターンも作成できる。また，三声それぞれの音色を自由に変えられる。メロディの入力方法は簡単で，誰にでもわかるものとなっており，修正も容易。作成した曲のディスプレイ表示，プリントアウトも可能だ。オリジナル曲を美しい譜面にしてくれる。定価は3,500円(テープ版・マニュアル付き)で，6月発売。

「デジ・ソング・エディター」は，ポピュラー・ヒット曲集とクラッシック名曲集の2種で，各々次の10曲が収められている。

ポピュラー：ライディーン，テクノポリス，シルクロード他

クラッシック：トルコ行進曲，G線上のアリア，アラベスク他

これら収録されている曲のオートプレイのほか，編集機能が付いており，テンポの変更，移調，転調，3声それぞれの音色の変更などができる。

また，「ミュージック・ハーモナイザー3」でつくったオリジナル曲のエディットも可能で，3声それぞれの譜面をCRT表示，プリント出力もできる。7月中旬発売予定で，定価は各3,000円(テープ版・マニュアル付き)。

マシン語特集partⅡ

4 マシン語プログラム実例集2

簡単なサブシステムの駆動法

吉田健司

6809CPUはきれいにまとまった命令セットと，たくさんのアドレッシングモードを持ち，かなり柔軟なプログラムを容易に書くことができます。

FMシリーズではI/Oの処理をBIOSで行っているため，ハードウェアが多少違っていてもBIOSの呼び出し方法が同じであれば，プログラムに互換性を持つことができます。マシン語でプログラムを作るときアセンブラを使うと楽なのですが，長いプログラムではメインメモリ上の32K弱のF-BASICがじゃまになります。だからといってオールRAMにするとBIOSもなくなってしまうので，I/Oルーチンを作成しなければなりません。

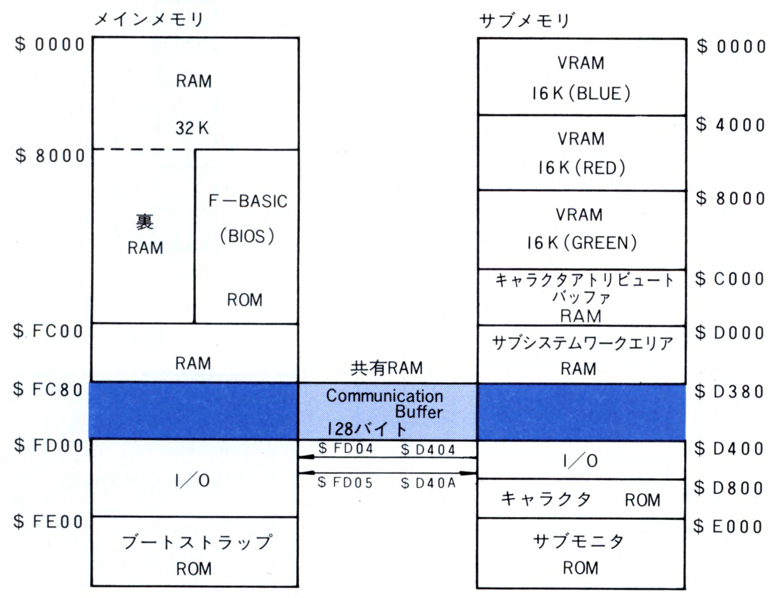
そこで，サブシステムを初めて使う人のために，オールRAMで使える汎用サブルーチンを紹介します。

サブシステム

富士通が発表したBIOSはFM-8，7，11ともほぼ同じで，またサブシステムとのデータのやりとりは**図1**のように，コミュニケーションバッファと呼ばれる共有RAM(128バイト）を通して行います。この共有RAMは，メインCPUとサブCPUが同時にアクセスしないように，メインメモリの$FD05番地のビット7をリード/ライトすることによって確認します。

サブCPUが持っているI/Oは，CRTディスプレイとキーボードとブザーです。FM-7ではキーボードのデータがメインCPUとサブCPUの両方につながっています。これらのI/Oとデータのやりとりをするためにたくさんのコマンドがありますが，それをメインCPUが共通RAMにセットするとサブCPUのモニタはコマンドを解読して作業をします。メインCPUはサブCPUが作業している間，次の処理を行うことができます。そこでたくさんのコマンドの中からキャラクタを扱うものを使ってみました。

図1　メモリマップ



初期設定

BASICのWIDTHとCONSOLE，COLOR文は，INITルーチンで行います。**図2**はその設定方法で

WIDTH　40,25
CONSOLE　0,25,0,0
COLOR　1

と同じです。私の趣味で背景色は青，文字は白がいちばん目が疲れにくいと思いますが，いかがでしょう(NHKのニュースもこれです)。

CONSOLEルーチンはカーソル表示やオーダ動作の設定を行います。オーダ動作をしないとCRとかLFなどの制御コードはそ

のままキャラクタとして表示されます(**図3**)。

図2 INITコマンド
(FM-7,8用)

$FC80	0	
1	0	
2	1	コマンドコード
3	1	背景色(0～7)
4	$28	横文字(80or40)
5	$19	行　数(25or20)
6	$00	スクロール開始行
7	$19	スクロール行数
8	$00	ファンクション表示(0:なし)
9	$FF	画面クリア(0:なし)
A	$00	グリーン文字(0:なし)

図3 CONSOLEコマンド
(FM-7,8用)

$FC80	0	
1	0	
2	$C	コマンドコード
3		bit 0～5下記参照

bit 0　カーソル表示(1:する, 0:なし)
bit 1　オーダ動作(1:する, 0:なし)
bit 2　TAB動作(1:する, 0:なし)
bit 3　ページウェイト(1:する, 0:なし)
bit 4　プットウェイト(1:する, 0:なし)
bit 5　CRコードでLF(1:する, 0:なし)

1文字出力

PUNCHルーチンはAレジスタの内容をCRTへ出力します(**図4-a**)。このルーチンは,1文字出力するたびにカーソル位置も自動的に移動します。また,カーソル移動キーやBS,DELなどの動作もします。PRTFLGの内容を0以外にして,1OUTルーチンをコールすると,画面と同時にプリンタへも出力します。

1文字入力

INKEYルーチンはキーボードからのデータを1文字分Aレジスタへ格納します。このルーチンはキーボードからデータが来るまで待っています。つまりNULLコード($00)は無視するわけです。画面表示は行いません。キーボードから受け取ったデータを画面へ出力するときは次のようにします。

```
  ⟆
LBSR INKEY
LBSR PUNCH
  ⟆
```

PUT

PUTルーチン(**図5**参照のこと)は先述のPUNCHルーチン同様,1文字出力を行いますが,カーソルは移動せず,またオーダ動作も行いません。

ワークエリアのPOSXにX座標,POSYにY座標のデータを,COLORにカラーコード,CHRAに表示させたいキャラクタコードを入れてコールすると,画面にキャラクタ1つを表示します。X,Y座標は**図6**のように0から始まります。

LOCATE

BASICのLOCATE文を真似て,画面の任意の位置に文字を表示するルーチンです。POSX,YにX,Y座標の値を,KAZUに文字数を,Uレジスタに文字列が格納されているアドレスを入れてコールします。

このルーチンはPUNCHルーチンと同じコマンドを使っています。カーソル表示とオーダ動作も行います(**図4-b**)。BASICと対比させてみると次のようになります。

○BASIC
```
100 LOCATE10,10,1
110 PRINT "Oh!FM"
```
○マシン語
```
     LDA   #10
```

図4 PUTコマンド

(a) 1文字出力

$FC80	0	
1	0	
2	3	コマンドコード
3	1	文字数
4	"A"	文字

(b) 文字列出力

$FC80	0	
1	0	
2	3	コマンドコード
3	8	文字数
4	SBA	出力文字
5	X	
6	Y	
7	SF	
8	at	
9	"A"	
A	"B	
B	"C"	
	⋮	

出力文字
↓

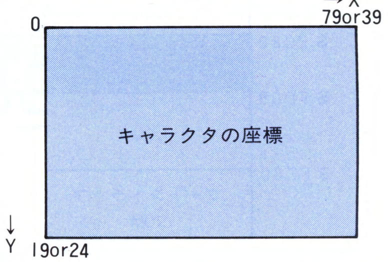
オーダ	オーダコード
SBA	$12
SF	$11
at	アトリビュート

即ちBASICで記述すると
LOCATE X, Y :COLOR 情報:PRINT "ABC ……"
となる

図5 PUT Character BLOCK 1コマンド

$FC80	0	
1	0	
2	7	コマンドコード
3		X1
4		Y1
5		X2
6		Y2
7		色コード
8		文字数
9		文字
A		文字

図6



```
        STA   POSX,PCR
        STA   POSY,PCR
        LDA   #5
        STA   KAZU
        LEAU  MOJI,PCR
        LBSR  LOCATE
          ∫
MOJI:FCC /Oh!FM/
```

サンプルプログラム

今まで説明した汎用サブルーチンを利用して，２つのプログラムを作ってみました。

サンプル１は，画面の中を黄色のキャラクタ＄87がはずんで動きまわります。これはPUTルーチンを利用しています。

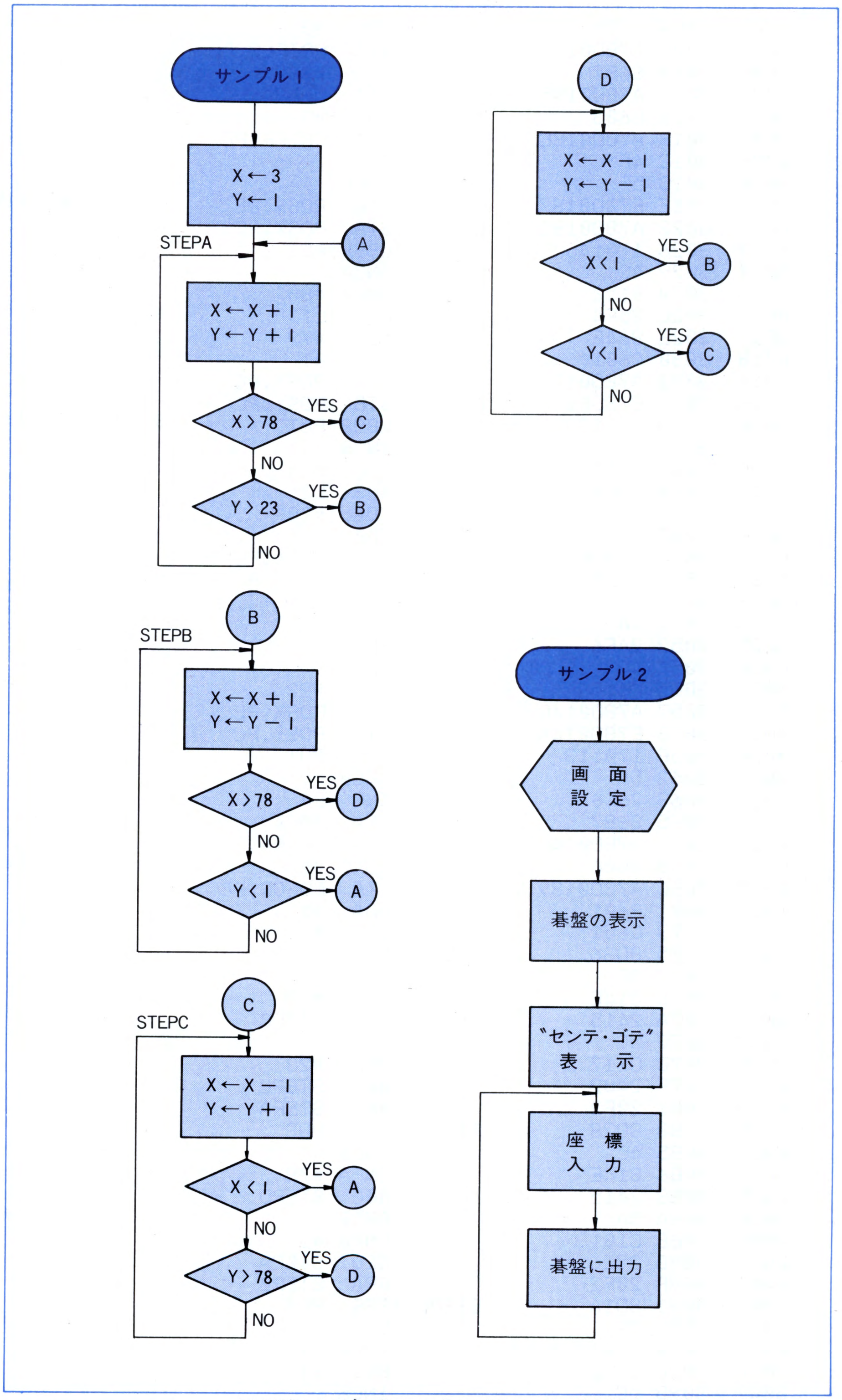
サンプル２は，LOCATEルーチンを利用した碁盤です。囲碁，五目ならべにでも使ってください。スペースキーを押すと再び始まります。これらはすべてリロケータブルでどのアドレスからもスタートできます。

汎用サブルーチンはメインプログラムの後ろに置き，また，メインの頭の部分では必ず，LDA，＃$FD，TFR　A，DP（$86，$FD，$1F，$8B）を設定してください。

また，サンプルプログラムから抜け出すには，STOP＋RESETとしてください。

おわりに

サブシステムとやりとりするコマンドの中には，グラフィック関係の命令などもあり，同じようにコマンドを送れば簡単に扱うことができます。6809CPUはZ80と違ってフラグ（CCR）がよく動く点が特徴でしょう。これによって条件判断がかなり楽になっていると思います。このCPUにI/Oマップを持たせると，さらに面白いと思います。紹介したサブルーチンはあまり整然としていませんが，サブシステムを扱うのに少しでも役に立てば幸いです。



○ここで扱われるプログラムは，個人で利用するほかは著作権法上無断複製を禁じられています。
COPY RIGHT © 1983 KENJI YOSHIDA

サンプルプログラム１

```
3000                    ORG   $3000
3000          ;
3000          ; sample1  (PUT CHARACTER BLOCK1)
3000          ;*** MAIN ROUTINE ***
3000 86FD     MAIN  :LDA   #$FD      } DPレジスタを$FDに設定
3002 1F8B             TFR   A,DP
3004 1700B2           LBSR  INIT     } 画面初期設定
3007 1700EC           LBSR  CONSOL
```

```
300A 8600                LDA   #0
300C A78D0197            STA   PRTFLG,PC     } プリンタ出力なし
3010 8687                LDA   #$87
3012 A78D0195            STA   CHARA,PC
3016 8603                LDA   #3
3018 A78D018E            STA   COLOR,PC
301C 4F                  CLRA
301D 5F                  CLRB
301E E78D0187            STB   POSY,PC       } キャラクタ"■"を紫色でY=0, Xを0から79まで表示
3022 A78D0182   L1      :STA   POSX,PC
3026 17014C              LBSR  PUT
3029 4C                  INCA
302A 8150                CMPA  #80
302C 25F4                BCS   L1
302E 864F                LDA   #79
3030 C601                LDB   #1
3032 A78D0172            STA   POSX,PC
3036 E78D016F   L2      :STB   POSY,PC
303A 170138              LBSR  PUT           } X=79, Yを1から24まで"■"を表示
303D 5C                  INCB
303E C119                CMPB  #25
3040 25F4                BCS   L2
3042 864E                LDA   #78
3044 C618                LDB   #24
3046 E78D015F            STB   POSY,PC
304A A78D015A   L3      :STA   POSX,PC       } Y=24, Xを78から0まで"■"を表示
304E 170124              LBSR  PUT
3051 4A                  DECA
3052 2AF6                BPL   L3
3054 4F                  CLRA
3055 C617                LDB   #23
3057 A78D014D            STA   POSX,PC
305B E78D014A   L4      :STB   POSY,PC       } X=0, Yを23から1まで"■"を表示
305F 170113              LBSR  PUT
3062 5A                  DECB
3063 26F6                BNE   L4
3065 8687                LDA   #$87
3067 A78D0140            STA   CHARA,PC
306B 8606                LDA   #6            キャラクタ"■"を画面内で反射させる。(黄色)
306D A78D0139            STA   COLOR,PC
3071 8601                LDA   #1
3073 C601                LDB   #1
3075 8D36       STEPA   :BSR   POS
3077 4C                  INCA
3078 814E                CMPA  #78
307A 2415                BCC   STEPC         } X=1, Y=1の状態よりXとYを1つずつ増加させて右下りの方向へ進む。
307C 5C                  INCB
307D C117                CMPB  #23
307F 2402                BCC   STEPB
3081 20F2                BRA   STEPA
3083 8D28       STEPB   :BSR   POS
3085 4C                  INCA
3086 814E                CMPA  #78
3088 2415                BCC   STEPD
308A 5A                  DECB                } Xを増加させ, Yを減少させて, 右上に進む
308B C101                CMPB  #1
308D 27E6                BEQ   STEPA
308F 20F2                BRA   STEPB
3091 8D1A       STEPC   :BSR   POS
3093 4A                  DECA
3094 8101                CMPA  #1
3096 27DD                BEQ   STEPA
3098 5C                  INCB                } Xを減少させ, Yを増加させて左下へ進む
3099 C117                CMPB  #23
309B 2402                BCC   STEPD
309D 20F2                BRA   STEPC
309F 8D0C       STEPD   :BSR   POS
30A1 4A                  DECA
30A2 8101                CMPA  #1
30A4 27DD                BEQ   STEPB
30A6 5A                  DECB                } XとYを減少させて左上に進む
30A7 C101                CMPB  #1
30A9 27E6                BEQ   STEPC
```

```
0052  30AB 20F2          BRA   STEPD
0053  30AD          ;
0054  30AD A78D00F7 POS  :STA   POSX,PC
0055  30B1 E78D00F4      STB   POSY,PC
0056  30B5 1700BD        LBSR  PUT
0057  30B8 39            RTS
```

0054～0056: Aレジスタの内容をPOSXへ Bレジスタの内容をPOSYへ ストアして，表示させる。

注)このあとに，サンプルプログラム2の行番号00BF(汎用サブルーチン)が続きます。

サンプルプログラム2

```
0001  3000               ORG   $3000
0002  3000          ;
0003  3000          ; sample2
0004  3000          ;*** MAIN ROUTINE ***
0005  3000 86FD     MAIN :LDA   #$FD
0006  3002 1F8B          TFR   A,DP
0007  3004 1701E8        LBSR  INIT
0008  3007 170222        LBSR  CONSOL
0009  300A 6F8D030C      CLR   PRTFLG,PC
000A  300E 8607          LDA   #7
000B  3010 A78D0309      STA   COLOR,PC
000C  3014 8698          LDA   #$98
000D  3016 A78D0304      STA   CHARA,PC
000E  301A 8605          LDA   #5
000F  301C A78D02FB      STA   POSX,PC
0010  3020 C605          LDB   #5
0011  3022 E78D02F6      STB   POSY,PC
0012  3026 170282        LBSR  PUT
0013  3029 8691          LDA   #$91
0014  302B A78D02EF      STA   CHARA,PC
0015  302F 8606          LDA   #6
0016  3031 C611          LDB   #17
0017  3033 A78D02E4 L1   :STA   POSX,PC
0018  3037 170271        LBSR  PUT
0019  303A 4C            INCA
001A  303B 5A            DECB
001B  303C 26F5          BNE   L1
001C  303E 8699          LDA   #$99
001D  3040 A78D02DA      STA   CHARA,PC
001E  3044 8617          LDA   #23
001F  3046 A78D02D1      STA   POSX,PC
0020  304A 17025E        LBSR  PUT
0021  304D C606          LDB   #6
0022  304F 8693     L2   :LDA   #$93
0023  3051 A78D02C9      STA   CHARA,PC
0024  3055 8605          LDA   #5
0025  3057 A78D02C0      STA   POSX,PC
0026  305B E78D02BD      STB   POSY,PC
0027  305F 170249        LBSR  PUT
0028  3062 868F          LDA   #$8F
0029  3064 A78D02B6      STA   CHARA,PC
002A  3068 8606          LDA   #6
002B  306A A78D02AD L3   :STA   POSX,PC
002C  306E 17023A        LBSR  PUT
002D  3071 4C            INCA
002E  3072 8116          CMPA  #22
002F  3074 23F4          BLS   L3
0030  3076 8692          LDA   #$92
0031  3078 A78D02A2      STA   CHARA,PC
0032  307C 8617          LDA   #23
0033  307E A78D0299      STA   POSX,PC
0034  3082 170226        LBSR  PUT
0035  3085 5C            INCB
0036  3086 C116          CMPB  #22
0037  3088 23C5          BLS   L2
0038  308A 869A          LDA   #$9A
0039  308C A78D028E      STA   CHARA,PC
003A  3090 8605          LDA   #5
003B  3092 A78D0285      STA   POSX,PC
003C  3096 C617          LDB   #23
003D  3098 E78D0280      STB   POSY,PC
003E  309C 17020C        LBSR  PUT
```

- 0005～0006: DPレジスタを$FDに設定
- 0007～0008: 画面の初期設定
- 0009: プリント出力なし
- 000A～: 碁盤の表示
- 000A～0012: 色，キャラクタ，ポジションX，Yを設定してPUTルーチンをコール（色…白(7)，キャラクタ…“┌”ポジションX=5，Y=5）
- 0013～001B: キャラクタ“┬”を17個，Y=5，Xを6～22まで表示
- 001C～0020: X=23，Y=5の位置にキャラクタ“┐”を1つ表示
- 0022～0027: X=5そしてBレジスタの内容がYの位置にキャラクタ“├”を表示
- 0028～002F: キャラクタ“┼”をX=6～22まで表示。Yの位置はBレジスタの内容をPOSYに格納されている。
- 0030～0034: キャラクタ“┤”をX=23の位置へ表示
- 0035～0037: Bレジスタの内容を1ふやして22以下のときはL2へもどる。
- 0038～003E: X=5，Y=23の位置へキャラクタ“└”を表示

```
309F 8690              LDA   #$90
30A1 A78D0279          STA   CHARA,PC
30A5 8606              LDA   #6
30A7 A78D0270     L4  :STA   POSX,PC
30AB 1701FD            LBSR  PUT
30AE 4C                INCA
30AF 8116              CMPA  #22
30B1 23F4              BLS   L4
```

キャラクタ"⊥"をY=23, Xを6～22の位置へ表示

```
30B3 869B              LDA   #$9B
30B5 A78D0265          STA   CHARA,PC
30B9 8617              LDA   #23
30BB A78D025C          STA   POSX,PC
30BF 1701E9            LBSR  PUT
```

X=23, Y=23の位置へキャラクタ"」"を表示

碁盤の座標表示

```
30C2 8641              LDA   #$41
30C4 C604              LDB   #4
30C6 E78D0252          STB   POSY,PC
30CA C605              LDB   #5
30CC A78D024E     L5  :STA   CHARA,PC
30D0 E78D0247          STB   POSX,PC
30D4 1701D4            LBSR  PUT
30D7 4C                INCA
30D8 5C                INCB
30D9 C117              CMPB  #23
30DB 23EF              BLS   L5
```

X=5～23, Y=4の位置にキャラクタA～Sを表示する。
(Aレジスタにキャラクタデータ, BレジスタにX位置を設定)

```
30DD 8641              LDA   #$41
30DF C604              LDB   #4
30E1 E78D0236          STB   POSX,PC
30E5 C605              LDB   #5
30E7 A78D0233     L6  :STA   CHARA,PC
30EB E78D022D          STB   POSY,PC
30EF 1701B9            LBSR  PUT
30F2 4C                INCA
30F3 5C                INCB
30F4 C117              CMPB  #23
30F6 23EF              BLS   L6
30F8              ;
```

X=4, Y=5～23の位置にキャラクタA～Sを表示する。
(Aレジスタにキャラクタデータ, BレジスタにY位置を設定)

打ち石座標などの表示

```
30F8 861A              LDA   #26
30FA A78D021D          STA   POSX,PC
30FE 8605              LDA   #5
3100 A78D0218          STA   POSY,PC
3104 338D00CF          LEAU  STR1,PC
3108 8608              LDA   #8
310A A78D0211          STA   KAZU,PC
310E 1701CC            LBSR  LOCATE
```

X=26, Y=5の位置からSTR1(●センテ)を表示
(POSX, POSY, KAZU, そしてUレジスタに表示データのあるアドレスを設定してLOCATEルーチンをコールする)

```
3111 860C              LDA   #12
3113 A78D0205          STA   POSY,PC
3117 338D00C4          LEAU  STR2,PC
311B 8606              LDA   #6
311D A78D01FE          STA   KAZU,PC
3121 1701B9            LBSR  LOCATE
```

X=26, Y=12の位置からSTR2(○ゴテ)を表示

```
3124 861B         L7  :LDA   #27
3126 A78D01F1          STA   POSX,PC
312A 8607              LDA   #7
312C A78D01EC          STA   POSY,PC
3130 8D72              BSR   ヨコ
```

X=27, Y=7の位置から"ヨコ?"と表示

```
3132 17008B            LBSR  INPUT
3135 A78D01E7          STA   BUFF,PC
```

キー入力を受け付けてデータをBUFFにストア

```
3139 861B              LDA   #27
313B A78D01DC          STA   POSX,PC
313F 8609              LDA   #9
3141 A78D01D7          STA   POSY,PC
3145 8D6B              BSR   タテ
```

X=27 Y=9の位置から"タテ?"と表示

```
3147 8D77              BSR   INPUT
3149 A78D01D4          STA   BUFF+1,PC
```

キー入力を受け付け, BUFF+1へストア

```
314D 86EC              LDA   #$EC
314F A78D01CB          STA   CHARA,PC
3153 8600              LDA   #0
3155 A78D01C4          STA   COLOR,PC
3159 8D32              BSR   POS
```

BUFFの内容をX, BUFF+1の内容をYとして"●"を表示

```
315B 861B              LDA   #27
315D A78D01BA          STA   POSX,PC
3161 860E              LDA   #14
3163 A78D01B5          STA   POSY,PC
3167 8D3B              BSR   ヨコ
```

X=27, Y=14の位置へ"ヨコ?"を表示

```
3169 8D55              BSR  INPUT
316B A78D01B1          STA  BUFF,PC
```
キー入力をBUFFへストア

```
316F 861B              LDA  #27
3171 A78D01A6          STA  POSX,PC
3175 8610              LDA  #16
3177 A78D01A1          STA  POSY,PC
317B 8D35              BSR  ﾀﾃ
```
X=27, Y=16の位置へ"タテ？"を表示

```
317D 8D41              BSR  INPUT
317F A78D019E          STA  BUFF+1,PC
```
キー入力をBUFF＋1へストア

```
3183 8607              LDA  #7
3185 A78D0194          STA  COLOR,PC
3189 8D02              BSR  POS
318B 2097              BRA  L7
```
指定した位置へ"○"を表示

```
318D            ;
318D 3412       POS    :PSHS A,X
318F 308D018D          LEAX BUFF,PC
3193 A680              LDA  ,X+
3195 A78D0182          STA  POSX,PC
3199 A684              LDA  ,X
319B A78D017D          STA  POSY,PC
319F 170109            LBSR PUT
31A2 3592              PULS A,X,PC
```
BUFFの内容をPOSXへ，BUFF＋1の内容をPOSYへストアして
PUTルーチンをコール

```
31A4 338D003D   ﾖｺ     :LEAU STR3,PC
31A8 8605              LDA  #5
31AA A78D0171          STA  KAZU,PC
31AE 17012C            LBSR LOCATE
31B1 39                RTS
```
STR3("ヨコ？")のデータのあるアドレスをUレジスタへ入れて，
LOCATEをコール

```
31B2 338D0034   ﾀﾃ     :LEAU STR4,PC
31B6 8605              LDA  #5
31B8 A78D0163          STA  KAZU,PC
31BC 17011E            LBSR LOCATE
31BF 39                RTS
```
STR4("タテ？")のデータのあるアドレスをUレジスタへ入れて，
LOCATEをコール

```
31C0 1700B8     INPUT  :LBSR INKEY
31C3 8120              CMPA #$20
31C5 1027FE37          LBEQ MAIN
31C9 8141              CMPA #$41
31CB 25F3              BCS  INPUT
31CD 8153              CMPA #$53
31CF 22EF              BHI  INPUT
31D1 17006C            LBSR 1OUT
31D4 803C              SUBA #$3C
31D6 39                RTS
```
キー入力を受け付けてキーの内容がA～Sのときは表示してリターン。
スペースのときは再び最初にもどる。その他は再びキー入力を受け付ける。

```
31D7            ;
31D7 1100       STR1   :FDB  $1100
31D9 EC                 FCB  $EC
31DA 1107               FDB  $1107
31DC BEDDC3             FCC  /ｾﾝﾃ/
```
SFオーダで色を黒。$ECは"●"
SFオーダで色を白，"センテ"を表示

```
31DF 1107       STR2   :FDB  $1107
31E1 EC                 FCB  $EC
31E2 BADEC3             FCC  /ｺﾞﾃ/
```
SFオーダで色を白，"○ゴテ"を表示

```
31E5 1107       STR3   :FDB  $1107
31E7 D6BA3F             FCC  /ﾖｺ?/
```
文字を白で"ヨコ？"を表示

```
31EA 1107       STR4   :FDB  $1107
31EC C0C33F             FCC  /ﾀﾃ?/
```
文字を白で"タテ？"を表示

```
31EF            ;
31EF            ;*** ﾊﾝﾖｳ SUBROUTIN ***
31EF            ;
31EF            BUSY   :EQU  $FD05      サブシステムとのインタフェース
31EF            PRTBSY:EQU   $FD02      ビット0がBUSY入力
31EF            PRTSTB:EQU   $FD00      ビット6がストローブ出力
31EF            PRTDAT:EQU   $FD01      プリンタへのデータ出力
31EF            COMUNI:EQU   $FC80      コミュニケーションバッファの先頭アドレス
31EF            ;
31EF 3414       INIT   :PSHS B,X
31F1 8D17              BSR  SUBHLT
31F3 308D0008          LEAX CODE1,PC
31F7 C60B              LDB  #11
31F9 8D22              BSR  STORE
31FB 8D1D              BSR  SUBST
31FD 3594              PULS B,X,PC
31FF 00000101   CODE1  :FDB  0,$0101
3203 28190019          FDB  $2819,$0019
3207 00FF              FDB  $00FF
3209 00                FCB  0
```
画面の初期設定
背　景　色　1
桁　　数　40($28)
行　　数　25($19)
スクロール始　0
スクロール数　25($19)
ファンクションキー表示なし
画面クリア
グリーン表示なし

```
320A                 ;
320A 3402            SUBHLT:PSHS A
320C 9605            RDY?  :LDA  <BUSY
320E 2BFC                   BMI  RDY?
3210 8680                   LDA  #$80
3212 9705                   STA  <BUSY
3214 9605            OK?   :LDA  <BUSY
3216 2AFC                   BPL  OK?
3218 3582                   PULS A,PC
```

サブCPUがレディまで待ち，$FD05のビット7を1にする。

```
321A                 ;
321A 0F05            SUBST :CLR  <BUSY
321C 39                     RTS
```

$FD05のビット7を0にして，サブシステムをスタートさせる。

```
321D                 ;
321D 3422            STORE :PSHS A,Y
321F 108EFC80               LDY  #COMUNI
3223 A680            TENSO :LDA  ,X+
3225 A7A0                   STA  ,Y+
3227 5A                     DECB
3228 26F9                   BNE  TENSO
322A 35A2                   PULS A,Y,PC
```

Xレジスタの示すアドレスのデータをYレジスタの示すアドレスへ転送する。
Bレジスタの内容は転送するバイト数

```
322C                 ;
322C 3414            CONSOL:PSHS B,X
322E 8DDA                   BSR  SUBHLT
3230 308D0008               LEAX CODEc,PC
3234 C604                   LDB  #4
3236 8DE5                   BSR  STORE
3238 8DE0                   BSR  SUBST
323A 3594                   PULS B,X,PC
323C 00000C03        CODEc :FDB  0,$0C03
```

カーソル，オーダなどの機能の設定
サブコマンド$C
（オートLFなし
PUTウェイトなし
ページウェイトなし
TABなし
オーダする
カーソル表示する）

```
3240                 ;
3240 3404            1OUT  :PSHS B
3242 8D0A                   BSR  PUNCH
3244 E68D00D2               LDB  PRTFLG,PC
3248 2702                   BEQ  NO
324A 8D1C                   BSR  PRINT
324C 3584            NO    :PULS B,PC
```

Aレジスタの内容を1文字だけ出力する。PRTFLGの内容が0でないときはプリンタにも出力

```
324E                 ;
324E 3434            PUNCH :PSHS B,X,Y
3250 8DB8                   BSR  SUBHLT
3252 308D000E               LEAX CODE3,PC
3256 C604                   LDB  #4
3258 8DC3                   BSR  STORE
325A 108EFC80               LDY  #COMUNI
325E A724                   STA  4,Y
3260 8DB8                   BSR  SUBST
3262 35B4                   PULS B,X,Y,PC
3264 00000301        CODE3 :FDB  0,$0301
```

Aレジスタの内容を1文字出力する。サブコマンド$3

```
3268                 ;
3268 3404            PRINT :PSHS B
326A D602            CHECK1:LDB  <PRTBSY
326C 54                     LSRB
326D 25FB                   BCS  CHECK1
326F 9701                   STA  <PRTDAT
3271 C600                   LDB  #0
3273 D700                   STB  <PRTSTB
3275 C640                   LDB  #$40
3277 D700                   STB  <PRTSTB
3279 3584                   PULS B,PC
```

プリンタ出力
PRTBSYのビット0が0になるまで待つ
Aレジスタの内容をプリンタへ
ストローブをL
ストローブをH

```
327B                 ;
327B 3404            INKEY :PSHS B
327D 8D8B            NUL   :BSR  SUBHLT
```

キー入力ルーチン

```
327F 308D0024               LEAX CODE29,PC
3283 C604                   LDB  #4
3285 8D96                   BSR  STORE
3287 8D91                   BSR  SUBST
```

コマンド$29をサブCPUへ転送

```
3289 17FF7E                 LBSR SUBHLT
328C B6FC84                 LDA  COMUNI+4
328F 2609                   BNE  W1
3291 C680                   LDB  #$80
3293 F7FC80                 STB  COMUNI
3296 8D82                   BSR  SUBST
3298 20E3                   BRA  NUL
```

$FC84の内容が0のときはキー入力なし。
キー入力なしの時は$00のコード(NULL)が$FC83へ現われる

```
329A B6FC83    W1    :LDA   COMUNI+3
329D C680            LDB   #$80
329F F7FC80          STB   COMUNI
32A2 17FF75          LBSR  SUBST
32A5 3584            PULS  B,PC
32A7 00002900  CODE29:FDB   0,$2900
32AB           ;
32AB 3436      PUT   :PSHS  A,B,X,Y
32AD 17FF5A          LBSR  SUBHLT
32B0 308D0026        LEAX  CODE7,PC
32B4 C603            LDB   #3
32B6 17FF64          LBSR  STORE
32B9 108EFC80        LDY   #COMUNI
32BD AE8D005A        LDX   POSX,PC
32C1 AF23            STX   3,Y
32C3 AF25            STX   5,Y
32C5 A68D0054        LDA   COLOR,PC
32C9 A727            STA   7,Y
32CB 8601            LDA   #1
32CD A728            STA   8,Y
32CF A68D004B        LDA   CHARA,PC
32D3 A729            STA   9,Y
32D5 17FF42          LBSR  SUBST
32D8 35B6            PULS  A,B,X,Y,PC
32DA 0000      CODE7 :FDB   0
32DC 07              FCB   7
32DD           ;
32DD 3436      LOCATE:PSHS  A,B,X,Y
32DF 17FF28          LBSR  SUBHLT
32E2 308D0031        LEAX  CODE3A,PC
32E6 C603            LDB   #3
32E8 17FF32          LBSR  STORE
32EB 108EFC80        LDY   #COMUNI
32EF A68D002C        LDA   KAZU,PC
32F3 8B03            ADDA  #3
32F5 A723            STA   3,Y
32F7 C612            LDB   #$12
32F9 E724            STB   4,Y
32FB E68D001C        LDB   POSX,PC
32FF E725            STB   5,Y
3301 E68D0017        LDB   POSY,PC
3305 E726            STB   6,Y
3307 8003            SUBA  #3
3309 3127            LEAY  7,Y
330B E6C0      J1    :LDB   ,U+
330D E7A0            STB   ,Y+
330F 4A              DECA
3310 26F9            BNE   J1
3312 17FF05          LBSR  SUBST
3315 35B6            PULS  A,B,X,Y,PC
3317 0000      CODE3A:FDB   0
3319 03              FCB   3
331A           ;
331A           ;*** WORK AREA ***
331A           PRTFLG:RMB   1
331B           POSX  :RMB   1
331C           POSY  :RMB   1
331D           COLOR :RMB   1
331E           CHARA :RMB   1
331F           KAZU  :RMB   1
3320           BUFF  :RMB   121
```

W1〜PULS B,PC: キー入力があった時は, $FC83の内容をAレジスタへ格納

PUT: PUT CHARACTER BLOCK I

LDY #COMUNI〜: コマンドの次にPOSX, POSY, COLOR, 転送バイト数(1つ), キャラクタを共有RAMへ送る。

LBSR SUBST: 実際は対角線座標として$(X_1, Y_1)-(X_2, Y_2)$を転送するが, このルーチンは$X_1=X_2$, $Y_1=Y_2$としてある。

LEAX CODE3A,PC: このサブルーチンはPUNCHルーチンと同じコマンド$3を使って任意の数の文字を出力する。

LDB #$12: オーダ動作のSBAを使って任意の位置へ表示する。

- PRTFLG — プリンタフラグ 0:プリントしない
- POSX — 表示位置 X
- POSY — 〃 Y
- COLOR — アトリビュート色指定
- CHARA — キャラクタコード
- KAZU — 文字数
- BUFF — キー入力用バッファ

※このアセンブラでは, PC相対アドレスモードでは下の例のようになります。

LDA OFFSET, PCR
⇩
LDA OFFSET, PC

サンプルプログラム1 ダンプリスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
3000	86	FD	1F	8B	17	00	B2	17	:0D
3008	00	EC	86	00	A7	8D	01	97	:3E
3010	86	87	A7	8D	01	95	86	03	:60
3018	A7	8D	01	8E	4F	5F	E7	8D	:E5
3020	01	87	A7	8D	01	82	17	01	:57
3028	4C	4C	81	50	25	F4	86	4F	:57
3030	C6	01	A7	8D	01	72	E7	8D	:E2
3038	01	6F	17	01	38	5C	C1	19	:F6
3040	25	F4	86	4E	C6	18	E7	8D	:3F
3048	01	5F	A7	8D	01	5A	17	01	:07
3050	24	4A	2A	F6	4F	C6	17	A7	:61
3058	8D	01	4D	E7	8D	01	4A	17	:B1
3060	01	13	5A	26	F6	86	87	A7	:3E
3068	8D	01	40	86	06	A7	8D	01	:8F
3070	39	86	01	C6	01	8D	36	4C	:96
3078	81	4E	24	15	5C	C1	17	24	:60
3080	02	20	F2	8D	28	4C	81	4E	:E4

3088	24	15	5A	C1	01	27	E6	20	:82
3090	F2	8D	1A	4A	81	01	27	DD	:69
3098	5C	C1	17	24	02	20	F2	8D	:F9
30A0	0C	4A	81	01	27	DD	5A	C1	:F7
30A8	01	27	E6	20	F2	A7	8D	00	:54
30B0	F7	E7	8D	00	F4	17	00	BD	:33
30B8	39	34	14	8D	17	30	8D	00	:E2
30C0	08	C6	0B	8D	22	8D	1D	35	:67
30C8	94	00	00	01	01	50	19	00	:FF
30D0	19	00	FF	00	34	02	96	05	:E9
30D8	2B	FC	86	80	97	05	96	05	:64
30E0	2A	FC	35	82	0F	05	39	34	:5E
30E8	22	10	8E	FC	80	A6	80	A7	:09
30F0	A0	5A	26	F9	35	A2	34	14	:38
30F8	8D	DA	30	8D	00	08	C6	04	:F6
3100	8D	E5	8D	E0	35	94	00	00	:A8
3108	0C	02	34	04	8D	0A	E6	8D	:50
3110	00	95	27	02	8D	1C	35	84	:20
3118	34	34	8D	B8	30	8D	00	0E	:78
3120	C6	04	8D	C3	10	8E	FC	80	:34
3128	A7	24	8D	B8	35	B4	00	00	:F9
3130	03	01	34	04	D6	02	54	25	:8D
3138	FB	97	01	C6	00	D7	00	C6	:F6
3140	40	D7	00	35	84	34	04	8D	:95
3148	8B	30	8D	00	24	C6	04	8D	:C3
3150	96	8D	91	17	FF	7E	B6	FC	:FA
3158	84	26	09	C6	80	F7	FC	80	:6C
3160	8D	82	20	E3	B6	FC	83	C6	:0D
3168	80	F7	FC	80	17	FF	75	35	:B3
3170	84	00	00	29	00	34	36	17	:2E
3178	FF	5A	30	8D	00	26	C6	03	:05
3180	17	FF	64	10	8E	FC	80	AE	:42
3188	8D	00	1D	AF	23	AF	25	A6	:F6
3190	8D	00	17	A7	27	86	01	A7	:A0
3198	28	A6	8D	00	0E	A7	29	17	:50
31A0	FF	42	35	B6	00	00	07	00	:33

サンプルプログラム2　ダンプリスト(FM-7,8用)

FM-IIの方は青字のように変更してください。

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
3000	86	FD	1F	8B	17	01	E8	17	:44
3008	02	22	6F	8D	03	0C	86	07	:BC
3010	A7	8D	03	09	86	98	A7	8D	:92
3018	03	04	86	05	A7	8D	02	FB	:C3
3020	C6	05	E7	8D	02	F6	17	02	:50
3028	82	86	91	A7	8D	02	EF	86	:44
3030	06	C6	11	A7	8D	02	E4	17	:0E
3038	02	71	4C	5A	26	F5	86	99	:53
3040	A7	8D	02	DA	86	17	A7	8D	:E1
3048	02	D1	17	02	5E	C6	06	86	:9C
3050	93	A7	8D	02	C9	86	05	A7	:C4
3058	8D	02	C0	E7	8D	02	BD	17	:99
3060	02	49	86	8F	A7	8D	02	B6	:4C
3068	86	06	A7	8D	02	AD	17	02	:88
3070	3A	4C	81	16	23	F4	86	92	:4C
3078	A7	8D	02	A2	86	17	A7	8D	:A9
3080	02	99	17	02	26	5C	C1	16	:0D
3088	23	C5	86	9A	A7	8D	02	8E	:CC
3090	86	05	A7	8D	02	85	C6	17	:23
3098	E7	8D	02	80	17	02	0C	86	:A1
30A0	90	A7	8D	02	79	86	06	A7	:72
30A8	8D	02	70	17	01	FD	4C	81	:E1
30B0	16	23	F4	86	9B	A7	8D	02	:84
30B8	65	86	17	A7	8D	02	5C	17	:AB
30C0	01	E9	86	41	C6	04	E7	8D	:EF
30C8	02	52	C6	05	A7	8D	02	4E	:A3
30D0	E7	8D	02	47	17	01	D4	4C	:F5
30D8	5C	C1	17	23	EF	86	41	C6	:D3
30E0	04	E7	8D	02	36	C6	05	A7	:22
30E8	8D	02	33	E7	8D	02	2D	17	:7C
30F0	01	B9	4C	5C	C1	17	23	EF	:4C
30F8	86	1A	A7	8D	02	1D	86	05	:7E
3100	A7	8D	02	18	33	8D	00	CF	:DD
3108	86	08	A7	8D	02	11	17	01	:ED
3110	CC	86	0C	A7	8D	02	05	33	:CC
3118	8D	00	C4	86	06	A7	8D	01	:12
3120	FE	17	01	B9	86	1B	A7	8D	:A4
3128	01	F1	86	07	A7	8D	01	EC	:A0
3130	8D	72	17	00	8B	A7	8D	01	:D6
3138	E7	86	1B	A7	8D	01	DC	86	:1F
3140	09	A7	8D	01	D7	8D	6B	8D	:9A
3148	77	A7	8D	01	D4	86	EC	A7	:99
3150	8D	01	CB	86	00	A7	8D	01	:14
3158	C4	8D	32	86	1B	A7	8D	01	:59
3160	BA	86	0E	A7	8D	01	B5	8D	:C5
3168	3B	8D	55	A7	8D	01	B1	86	:89
3170	1B	A7	8D	01	A6	86	10	A7	:33
3178	8D	01	A1	8D	35	8D	41	A7	:66
3180	8D	01	9E	86	07	A7	8D	01	:EE
3188	94	8D	02	20	97	34	12	30	:50
3190	8D	01	8D	A6	80	A7	8D	01	:76
3198	82	A6	84	A7	8D	01	7D	17	:75
31A0	01	09	35	92	33	8D	00	3D	:CE
31A8	86	05	A7	8D	01	71	17	01	:49
31B0	2C	39	33	8D	00	34	86	05	:E4
31B8	A7	8D	01	63	17	01	1E	39	:07
31C0	17	00	B8	81	20	10	27	FE	:A5
31C8	37	81	41	25	F3	81	53	22	:07
31D0	EF	17	00	6C	80	3C	39	11	:78
31D8	00	EC	11	07	BE	DD	C3	11	:73
31E0	07	EC	BA	DE	C3	11	07	D6	:3C
31E8	BA	3F	11	07	C0	C3	3F	34	:07
31F0	14	8D	17	30	8D	00	08	C6	:43
31F8	0B	8D	22	8D	1D	35	94	00	:2D
3200	00	01	01	28	19	00	19	00	:5C
3208	FF	00	34	02	96	05	2B	FC	:F7
3210	86	80	97	05	96	05	2A	FC	:63
3218	35	82	0F	05	39	34	22	10	:6A
3220	8E	FC	80	A6	80	A7	A0	5A	:D1
3228	26	F9	35	A2	34	14	8D	DA	:A5
3230	30	8D	00	08 (06)	C6	04 (06)	8D	E5	:01
3238	8D	E0	35	94	00 (0D)	00	0C (01)	03 (07)	:45 (4B)
3240	34	04	8D	0A	E6	8D	00	D2	:14
3248	27	02	8D	1C	35	84	34	34	:F3
3250	8D	B8	30	8D	00	0E	C6	04	:DA
3258	8D	C3	10	8E	FC	80	A7	24	:35
3260	8D	B8	35	B4	00	00	03	01	:32
3268	34	04	D6	02	54	25	FB	97	:1B
3270	01	C6	00	D7	00	C6	40	D7	:7B
3278	00	35	84	34	04	8D	8B	30	:39
3280	8D	00	24	C6	04	8D	96	8D	:2B
3288	91	17	FF	7E	B6	FC	84	26	:81
3290	09	C6	80	F7	FC	80	8D	82	:D1
3298	20	E3	B6	FC	83	C6	80	F7	:75
32A0	FC	80	17	FF	75	35	84	00	:C0
32A8	00	29	00	34	36	17	FF	5A	:03
32B0	30	8D	00	26	C6	03	17	FF	:C2
32B8	64	10	8E	FC	80	AE	8D	00	:B9
32C0	5A	AF	23	AF	25	A6	8D	00	:33
32C8	54	A7	27	86	01	A7	28	A6	:1E
32D0	8D	00	4B	A7	29	17	FF	42	:00
32D8	35	B6	00	00	07	34	36	17	:73
32E0	FF	28	30	8D	00	31	C6	03	:DE
32E8	17	FF	32	10	8E	FC	80	A6	:08
32F0	8D	00	2C	8B	03	A7	23	C6	:D7
32F8	12	E7	24	E6	8D	00	1C	E7	:93
3300	25	E6	8D	00	17	E7	26	80	:3C
3308	03	31	27	E6	C0	E7	A0	4A	:D2
3310	26	F9	17	FF	05	35	B6	00	:25
3318	00	03	00	00	00	00	00	00	:03

マシン語特集partII

8ビット 5 CPU談義

究極の8ビットCPU6809とZ80

真＆文彦＋α

僕ら自慢のFMシリーズ，何かにつけて偉い，偉い，こりゃすごいと思っていた。なぜかって言えば，CPUに6809が搭載されているからで，とにかくこれは大変なことなんだ。6809をCPUに使ったマイコンってほかにあまりなくて，マイコン界で幅を利かせているNECやシャープのには全くない。だから偉いんだ，と自慢してたら，とんでもない奴が「Z80の方が偉い」なんて言い出した。8ビットCPUの中では6809が一番偉いと思っていたのに，こんなことを言われて非常に怒ってしまう。というわけで，こいつと口論せざるを得なくなってしまった。我が愛する6809の名誉を守るためだもの。

さて，そこでこの内容を今回発表することにしました。初心者というか，マイコンはじめて間もない読者の方にもわかってもらえるように，CPUの説明なんかもします。ま，軽い気持ちで読んでみてください。

「歴史的背景」なんてつまらない？

そもそも，今をさかのぼること6,7年前，マイコンと言ったら基板にICの類がどかどか載っていて「いかにも」メカニックという感じのものだった。メーカー製のパソコンなど当然発売されておらず，海の向こうでPETとかアップルなんて名前のマイコンが発表されたとかされないとか，そんな情報が飛びかっていた程度だった。日本では，NECがTK-80というやはり1枚の基板上にICと少しのキースイッチ，発光ダイオードが付いている，たったそれだけのものがキットの形で10万円近くで販売されていた。でもあまりの高価格と，その頃マイコンに興味を持っていた人がほとんどハードウェア畑から出てきたこと，自作した方が自分の好きなシステムが得られることなどの理由から，自分で作ってしまう人が多かったんだ。

で，その頃簡単に手に入るCPUといえば8080と6800とSC／MP（スキャンプ）だった。自分でマイコンを作るっていうことになれば，使うCPUも自分で選べるわけで，選ぶとなれば人間やっぱり迷ってしまう。そこで悩んだ人たちや，自分で選んだCPUをほめちぎり自分の正当性を主張する人達が，CPU論争をはじめた。

この頃は，8080vs6800の論争だったんだが，その後，80系（ハチマル）対68系（ロクハチ）と言われるようになった。8080の上位CPUとしてZ80，8085が，6800の上位というかファミリCPUとして6801，6802，6803，6805，6808，6809が出たために，一概に8080，6800と言えなくなってしまったことによるんだ。

この8080，アメリカのインテル社が出したCPUで，マイコンの初期には非常によく使われた。あのインベーダブーム，覚えている？ あのマシンにはこの8080が使われていたんだ。

片や6800，やっぱりアメリカのモトローラ社が出したんだが，発表が8080よりも遅れてしまったためにイマイチ普及しなかった。ここがすごく大きな点で，今だに尾をひいている。実際，今でも80系であるZ80を使ったパソコンの方が，68系のCPUを使ったパソコンより多い。

このほかにもう一つメジャーなCPUがあって，6502というんだけれど，これはMOS（モス）テクノロジーっていう会社から発表された。PETとかアップルなんかで使っていて，特徴はとにかく速いこと。命令とかハードウェアは6800に似ていて，だからこのCPUも68系の中に入れていることも多い。

で，今回は6809vsZ80

以前行われていた論争，当事者がバカらしくなってやめたみたい。その後，パソコンが普及しはじめた頃，再び68系対80系論争がおこった。でもカシコい人たちが既に加わらなくなっていたため，わけのわからない言い合いに終始して，それも立ち消えとなった。

そこで今回，6809とZ80。当たり前のことだけど，やっぱり6809の方が偉いよねえ。

いよいよ本論

文彦　僕，PC-88のユーザーです。よろしく。何と言ってもZ80はすごくて，やっぱりZ80がいいと思います。だからZ80のため精一杯がんばります。

真　僕は今，お寿司が食べたいんですが，この談義が終われば，編集の人が食べさせてくれるというので，一生懸命09（マルキュー）の偉大さ

を主張します。やはりCPUは6809のサビ抜きの握りに限りますね。

Zはレジスタが多いんだぜ

文彦 何が偉いって，Z80はレジスタ[*1]が多い。**表1**を見てください。8ビットのレジスタが18個，16ビットが4本もある。8ビットレジスタのBC，DE，HLはペアにして16ビットとしても使えるから，16ビットのレジスタが10本あるとも言える。プログラムカウンタ[*2]はちょっと性質が違うから除いても，まだ9本もある。レジスタを使うのは，外部のメモリを使うより速いから，やっぱり多い方がいいんだ。

真 確かにレジスタの総数はZ80と比べて少ないけれど，Z80のレジスタってそう使いやすいとは思わないな。だって裏レジスタと呼ばれている8個のレジスタは，表レジスタ使用中はデータを憶えていても値は読み出せないじゃないか。読んだり書いたりするたびにレジスタの内容を交換しなければならない。

文彦 やっ，ま，そういう見方もあるけれど。**表1**で，メインレジスタと書いてあるのが通称「おもて」レジスタで，オルタネートが「うら」レジスタなんだよね。確かにH'（ダッシュを付けると裏レジスタを示す）をAに転送するのは，一筋なわではいかない。HからAに転送するのは

```
        LD      A,H
```

でいいけど，H'からAへは，

```
        EX      AF,AF'
        EXX
        LD      A,H
        EXX
```

となる。でもいいんじゃない，こんな用途はあまりないんだから。自由度は小さくても十分便利だぜ。

表1 Z80のレジスタ

メインレジスタセット		オルタネートレジスタセット	
A（アキュムレータ）	F（フラグレジスタ）	A'（アキュムレータ）	F'（フラグレジスタ）
B（汎用レジスタ）	C（汎用レジスタ）	B'（汎用レジスタ）	C'（汎用レジスタ）
D 〃	E 〃	D' 〃	E' 〃
H 〃	L 〃	H' 〃	L' 〃
←8ビット→			

←16ビット→	
IX（インデックスレジスタ）	
IY（ 〃 ）	
SP（スタックポインタ）	
PC（プログラムカウンタ）	
I（インタラプトページアドレスレジスタ）	R（メモリリフレッシュレジスタ）
←8ビット→	

HLしか使えない

真 それだけじゃない。例えばHLレジスタ[*3]なんて，このレジスタを使わなくては困るという用途が多すぎるんだよね。メモリの内容を読み出すとき，Z80だとHLレジスタを使う場合が多いんだ。また16ビットの演算にHLレジスタは不可欠ときている。その点09は，数値演算はDレジスタ，インデックス[*4]にはX，Y，Uレジスタを使えば問題ない。だいたいZ80はレジスタ間の演算は楽だけど，いったんメモリを使うと不便だよ。09だとレジスタがZ80より少ない分以上にアドレッシングモード[*5]が豊富でラクチン，ラクチン。つまりレジスタだけ使えばいいような処理ならZ80に分があるけど，こういうケースはそう多くなくて，やっぱりメモリをひんぱんに使うことになっちゃう。こうなると09の天下！

文彦 だけどHLの代わりにIXもIYも使えるよ！IX，IYという2本の16ビットレジスタがあって，この2本はだいたいHLレジスタのように使えるんだ。ま，少しはHLレジスタにしかできないこともあるけれど。

表2 6809のレジスタ

←16ビット→	
X-インデックスレジスタ	
Y-インデックスレジスタ	
U-ユーザースタックポインタ	
S-システムスタックポインタ	
PC-プログラムカウンタ	
アキュムレータA	アキュムレータB
D	

←8ビット→
DP(ダイレクトページレジスタ)
CC(コンディションコードレジスタ)

アドレッシングモードは悪い？

真 まあ，いろいろ使えるというのはそれだけ便利にできていると言えるけどね。でもIX，IYといったインデックスレジスタって全然強力じゃないね。09だとオフセット[*6]は定数でもアキュムレータ[*7]でもいいし，オフセットも最高で16ビット。それに比べてZは8ビット。おまけに09はインデックストインダイレクト[*8]もある。たとえ

[*1]**レジスタ**：CPUの内容にある記憶場所，CPUの計算はここを中心に行われる。通常，8ビットか16ビットの大きさを持つ。
[*2]**プログラムカウンタ**：実行中のアドレスが入る。当然，命令を実行するたびに書き替わる。そのため，記憶などには絶対使えない。
[*3]**HLレジスタ**：HレジスタとLレジスタを結合して，HLレジスタとして使える。H，Lとも8ビットのレジスタだから16ビットレジスタになる。
[*4]**インデックスレジスタ**：16ビットのレジスタで，データのあるアドレスを示すことが多い。そのためにインデックス（索引）と呼ばれている。Z80ではIX，IYがインデックスレジスタである。
[*5]**アドレッシングモード**：演算や転送の対象を示す，示し方のことを言う。
[*6]**オフセット(offset)**：㊇相殺するもの。埋め合わせ。何か主となるものに付け加えたものをオフセットという。アドレッシングモードにインデックスモードというのがあり，これはインデックスレジスタが指し示すアドレスが処理対象となる。ここでは，インデックスレジスタと定数を足したものや，アキュムレータと呼ばれるレジスタの内容を足したものが指し示すアドレスを処理対象とすることができるということ。

ば09で

```
LDA    〔9,X〕
```

が，Zでは

```
LD    H,(IX+10)
LD    L,(IX+9)
LD    A,(HL)
```

としなければならない。これは8ビットオフセットの例だけれど，16ビットだと(IX+d)といったものも通用しない。こういうメモリの読み出し方は，そうひんぱんに使うわけじゃないけど，こういうことが簡単にできる・できないは，プログラミングに大きな差を生むんじゃない？

文彦　それはそうだけど……。そもそもだね，68系のイヤなところは，このアドレッシングモードにある。なんでそんな面倒なことを覚えなければいけないんだよ。なんで，LDA #$10，LDA >$10，LDA <$10[9] の3つそれぞれ意味が違うんだ？ こうなると非人間的な記憶力を要求している，とまで言いたくなる。Z80だったら，

```
LD    A,80H
```
[10]

がイミディエイトだし，

```
LD    A,(80H)
```
[11]

ならエクステンドだ。Zはカッコが付けば，その番地の内容を示すんだ。なんとわかりやすいアセンブリ言語だろう。全く，人間的な優しさに満ちあふれているではありませんか。え，どうだ，何とか言ってみろ。

真　別にいーじゃないか，#がイミディエイトの印なんだよ。つまり09だとオペランド(操作の対象を示す部分)はアドレスを書くっていうだけじゃないか。たとえば先の

```
LDA    〔9,X〕
```

を，カッコで囲ってアドレスという表し方だと

```
LDA    ((X+9))
```

となって，みにくくなるじゃないか。だいたいレジスタ-メモリ間の命令が多いのにそのたびにカッコなんて……。

文彦　みにくくなるようなら，いらない。インダイレクトなんか，ない方がましだもんね，こんなアドレッシングモードを覚えなければならないんじゃね。

できない者のヒガミ?!

真　それはできないやつのヒガミだ。これ

[7]**アキュムレータ**：レジスタの一種で，計算の中心となるところ。演算器と訳される。
[8]**インデックストインダイレクト**：やはりアドレッシングモードの一種。インデックスレジスタが指し示すアドレスのデータをアドレスとして，処理対象を決める。
[9]LDA　#$10はイミディエイト，LDA　>$10はエクステンド，LDA　<$10はダイレクト，の各アドレッシングモード。
[10]LD　A，80HではAレジスタに16進の80が入る。
[11]LD　A，(80H)ではAレジスタに，80番地に格納されているデータが入る。

CPUって何だ！

CPUって何だろう。パソコンに新製品登場っていうと，まず，CPUに何を使っているのか気にする人が多いところからすると，やっぱり大事な部分なんだな，って思ってしまう。もちろんパソコンがCPUだけでできてるわけじゃないから，パソコンの良し悪しがCPUで決まるわけじゃないんだけれど，気になるのはやっぱりそれなりの理由があるんだ。

コンピュータの最も基本的な動作，つまり命令を実行するのがCPUなんだ。コンピュータの命令というとBASICを思い浮かべる人も多いんじゃないかと思うけど，ここで言う命令とは，もっと符号化された命令のこと。BASICはこの命令(実はマシン語のこと）が解読して実行するものだ，っていうことはみんな知ってるよね。……ここまでは知ってる人が多いと思うけど，問題はこの先。命令って，一体どのようにCPUに与えられるんだろう。また，処理されるデータは，一体どこに入っているんだろう。

実は命令もデータも，メモリっていう記憶装置に入っていて，CPUが必要に応じて取ってくる。そして，処理の結果もCPUがこのメモリにしまうわけ。これだけでどうしてパソコンは画面表示やキー入力などできるのかというと，これらもメモリの内容を取ってきたり，メモリにしまったりする操作のバリエーション（変形）として行えるからなんだ。

CPUの仕事はだいたいこんなもんだけど，メモリ上のデータのやりとりはどのように実現されるんだろう，と疑問。だって，メモリっていろいろなものを憶えておくのだから，どのデータがほしいのかはっきりさせなきゃ，困ります。

そこでデータ(命令も含めて)が入る場所に番号を付けることにする。これをアドレス(番地)って呼ぶんだ。「メモリの××番地には○○のデータが入っている」っていう感じ。さて，1つの番地にどのくらいデータが入るんだろう。僕たちが普通使うCPUでは，1つの番地に8ビットのデータが入る。ビットっていうのは「2とおりのものが表せる」情報で，これが8個ある，というのは2とおりが8個，つまり「256とおりのものが表せる」っていうこと。8ビットCPUっていうのは，この大きさの情報を一度に処理するのが基本のCPUなんだ。

ICではこれらがディジタルの電気信号として扱われるけど，具体的にはどうなっているんだろうか。まず，ディジタルの電気信号だけど，1つの信号で2とおりのことを表現する。すると，先ほどのデータを出し入れするためには，信号線が8本あればいいわけだ。アドレスの方はどうだろう。××番地の××を2進数16桁で表すのが今の8ビットCPUの標準だから，(2進数16桁)＝(16本の信号線）ということで，アドレスを示す信号は16本となる。なぜアドレスがこの大きさなのか，っていうと，大きさがなかなかよいことと，データ2つでちょうど表せることからなんだ。

ほかにもCPUには，データをメモリに送るのかメモリから受け取るのか表す信号や，さまざまなタイミングをとるためのクロック信号などがある。クロックっていうのは号令みたいなもので，クロックを速くする（＝号令を速くかける）と実行も速くなる，というわけなんだ。

を使うと，エレガントなプログラムになることが多いんだ。とくに09はスタックポインタもインデックスレジスタとして使えるから，アドレスを表す数値をスタックに入れてあるとき，これを使う*[12]と，いったんレジスタに目的のアドレスをロードする必要がないもんね。

文彦　んっなんだ，なんだ。オートインクリメント*[13]がなんだってんだ。スタックに数値のアドレスを置いたら

```
POP   HL
LD    A,(HL)
INC   HL
INC   HL
```

で，いいじゃないか。つまりだなあ，ややこしいアドレッシングモードを持つよりも，シンプルで理解しやすい命令で記述すればいいんで，その方がきれいなリストができるんだ。文句あるか！

真　あるとも！　いったんわかってしまえば，１命令で済むことが２命令になるなんてバカバカしい。リストもその分長くなるし，レジスタは１個むだになる。確かにインダイレクトを使うと１命令でバイト数はかかるよ。だけどアドレッシングモードが多いのは，その分柔軟性に富んでいることになるし，そもそもメモリを参照する命令のほとんどどれにも，様々なアドレッシングモードが同じように使えるって最高さ！

6809のアセンブリ言語が好き

文彦　6809のアセンブリ言語が気にいらん。なんで，

```
MOV   A,〔X+3〕
```

じゃなくて，

```
LDA   〔3,X〕
```

になるんだ。オペコード*[14]の中にオペランド*[15]を含むなんて，絶対に許せん。その点，Zはきれいなんだな。転送に関する命令は全てLDだし，ジャンプはJPがエクステンドでJRがリラティブ。6809なんか，同じブランチ*[16]なのに，BLOとかBHSとか，もうほとんどわけがわからん。やっぱり優しさに欠けてるよね。だからZ80のアセンブリの方がよっぽどいい。

真　確かに転送命令はZ80がきれいだ。だけど，ありそうでないというのがたくさんあるじゃないか。たとえば，

```
LD    BC,(HL)
```

なんてできないし。こういう例外をいちいち憶えていたらキリないよ。それに条件分岐だって，B××（××が条件）と表すのがなんでいけないんだ。別にニモニック*[17]

*[12]初心者の方は，もうほとんどワヤクチャで，何がなんだかさっぱりわからないことだろう。読み飛ばしてください。ちなみに，ここでは，LDA〔,S++〕のことを言っている。
*[13]**オートインクリメント**：LDA〔,S++〕でSの次の++がオートインクリメント。この命令でシステムスタックSがインクリメント（増加）されることを指している。
*[14]**オペコード**：命令のこと。
*[15]**オペランド**：命令がデータとして使う部分。BASICでたとえていうと，PRINT〝A〞ではPRINTがオペコード，〝A〞がオペランド。GOTO100ではGOTOがオペコード，100がオペランド。
*[16]**ブランチ**：枝のこと。GOTOと同じで，ジャンプのことをいう。

なぜCPUが何種類もあるの？

CPUの機能というのは，データを処理すること。記憶装置からデータを持ってきてそれに何らかの処理をして，また記憶装置にしまいこむ。外部の何らかの装置を介していろいろな情報を取り込み，その結果，外部の装置を動かす。こういった動作をするのにかわりはないのに，CPUはみんなこんな動作をしているのに，何種類ものCPUが存在しているんだ。おかしいと思う人もいるんじゃないかな。でも，よく考えてみると当たり前のことだよね。冷蔵庫もテレビ，机，服，人も，みんなそれぞれ同じ機能なのにたくさんの種類がある。これと同じなんだな。

たとえばテープレコーダ。機能としては，音を記録して再生するだけなのに，ほんとにいろいろなものがあるよね。ウォークマンタイプのテープレコーダ，ラジカセ，デッキなんて種類があって，その中でもステレオとかモノラルとか，再生しかできない，安い，高い，なんて区別がある。ほんとは音質が良くて，ステレオで録音できて，軽くて，小さくて，安くて，見た目が良くて，消費電力が小さければそれ一台で済むんだろうけど，なかなかそんなものは作れない。

CPUにおいても然り。どんな用途でも使いやすくて，速くて，値段が安ければそれ一種類があれば済むんだけど，まだそんなものはできていない。というより，できないんだ。なぜかな？

まず，CPUが8ビットだということ。8ビットっていうことは，処理単位が2^8，つまり256になるんだ。この8ビットを1バイトと言うんだけれども，256とおりしか1バイトでは表現できない。単純に考えれば1バイトでは256とおりの命令しか作れなくて，これでは不充分なんだよね。かといって命令が1バイト以上，つまり1処理単位以上におよぶと実行に時間がかかる。それから集積度の問題。LSIの中心にあるチップに電気回路が組まれてるんだけど，回路で使ってる素子（部品のことだと思って）の数が限られてしまうんだ。また，回路によって違うけど，一般的には素子の数が増えれば増えるほど，実行が遅くなってくる。この2つの制限が大きな要因となって，理想的な8ビットCPUができないわけ。

そこで，制限の枠中で使いやすいCPUを作るために，得意な分野を決めるなどしている。ここに複数のCPUが存在する必要があるんだ。機械に組み込んで制御用に使うCPUと，キーボードをつけて本来のコンピュータらしい使い方をするものの2とおりに，分野は大きく分けられる。で，その分野の中でも，どのような命令体系を持つか，どんなハードウェアを持つかで，またさらに分かれる。といったわけなんだ。これがCPUが何種類もある理由なんだよ。

の中に，条件が入っていたっていいじゃないか。そんなもの気分的なものさ。そもそもＺ80の条件分岐の種類が少ないから，そういうことを言いだすんだ。たとえば8000番地の内容とアキュムレータＡを比較して大小関係で分岐するとき，考えられるのは＞，≧，＝，≦，＜，≠の６とおりあるのに，Ｚ80では４とおりの分岐しかできない。つまり，

```
LD    HL,8000H
CP    (HL)
JP    (条件),(飛び先)
```

で，この（条件）には，Ｚ，NZ，Ｃ，NC の４つ，つまりＡ＝(8000H)，≠，＜，≧に対応するものはあるけれど，残りの２つがこれではできない*18。２回分岐命令を使わないとダメだ。09だと，

```
CMPA  $8000
Bcc   (飛び先との相対アドレス)
```

で，ccをEQ，NE，HI，HS，LS，LOとすれば，それぞれ＝，≠，＞，≧，≦，＜に対応する。まあ，これが８ビット符号なしだからまだ楽だけど，これが16ビットになったり符号付きになったりすると，Ｚ80では面倒でやりきれない。

文彦　いや，そうじゃなくて。そうじゃなくてですね，命令がわかりにくいと言っているんですよ。なんで

```
BRA   HS,(飛び先)
```

にならないの？　なんで

```
BSEQ  SUBPRO
```

みたいに，条件付きサブルーチンジャンプがないの？　気分的とかの問題じゃなくて，オペコードの中にオペランドが入るなんて絶対許せん。わかりにくい。困る。何が「究極の８ビット」だ！　俺たちにはＺ800*19シリーズという強い味方も控えているんだ。

真　確かに条件によってサブルーチンコールするのは，命令一つじゃできないけれど……。僕はこういう命令の書き方，そう悪いとは思わない。そんなことで究極の８ビットでなくなるものではない。だいたいＺ800って本当に出るの？

文彦　出ます，出るんです。そして，これが真の「究極の８ビット」となることでしょう。結局，80系がいいんです。僕ら80系にはソフトの蓄積がある。6809にCP/M*20のようなソフトがありますか？　絶対数で見ればＺ80の方が圧倒的に多いのだから。ソフトの蓄積は，ユーザーにとってみればかなり大きな問題だ。それがために消えていったCPUも多い。今，生き残っている８ビットCPUは，Z80，6502，6809で，みんなある程度のソフトは持っているけれど，やっぱりZ80がダントツだからね。

命令体系なら6809

真　何がＺ800だ。結局のところ，最初のマイコンがＺの先祖8080しかなかったからじゃないか。君は細かい所をずい分つっついてきたけど，命令の体系は09の方が整っている。とは言っても，6809の先祖6800の名残りで，指摘された以外にも気持ち悪い点もあるけど。でもやっぱりプログラムは09の方が作りやすい。Ｚ80だとちょっとのことでも命令数がすぐ増えてしまう。実行速度や命令長は何とも言えないけれど。

要するに，先に80系が出たことに諸悪の根源があって，その遺産でＺ80のソフトが多いだけだ。現行ソフトを使うならＺ80もいいけど，作るならやっぱり09と言いたいね。

文彦　確かに命令体系はね。うん，はっきり言ってね。とても，くやしいけれどね。とは言え，やはりCP/Mだな。68000がCP/M-68K*21を出したのがいい証拠だ。そのうちCP/M-09*22が出るに違いない。09にFORTRAN*23があるか？　COBOLがあるか？　LISPがあるか？

真　FORTRAN，PASCALぐらいFLEX*24にも……，だけどかなり見劣りするな。でもOS-9*25みたいなマルチユーザー・マルチタスクのＯＳは，Ｚ80ではちょっときついだろ？　この辺りに09の底力がある。ただユーザーがＺ80より少ないから……う～ん残念！　結局，今のところはCPU自体のおもしろさ，使いやすさを楽しむなら09，高級言語とかいろいろユーティリティを楽しむならＺ80になるのかな。

文彦　おまえ，だんだん冷静になってきたな。こういう論争っていうのは，お互い熱くなるからおもしろいんだ。

そりゃ冷静になれば，CPU自体は6809の方が圧倒的にいい。アドレッシングが強いというのは，本当に良くて，リエントラント*26にするためスタックを作ろうと思えば，絶対に09だよね。このごろはやりのアルゴリズムを使うなら09なんだな。ただアセンブリ言語が汚ないというのはあると思うんだ。

真　何度も言ってるけど，僕はあれはあれで結構いいと思う。ま，Ｚ80慣れした人には気持ち悪い面もあるだろうけれど，僕にはまとまって見える。

文彦　まとまって見えるのは，いいけれど。あ，真君，どこ行くの？　マサミチ君もターちゃんも，あ，もしかして寿司屋？　僕も行きたい。あ，行っちゃった。あ～あ。

6809とＺ80の対決，みんなまとまっていなくなったので，僕が感じるまま，ちょっとまとめてみよう。

そこで最後に

命令自体は，やっぱり6809がいいと思うね。スタックが２本もあるし，インデック

---

*17**ニモニック**：アセンブリ言語の命令語。
*18Z80では，ジャンプの条件として，Z，NZ，C，NC，P，M，PO，PEの８種類があるが，相対ジャンプでは，Z，NZ，C，NCの４種。
*19**Z800**：80系としては，８ビットの究極として期待されるCPU。ザイログ社が発表。特権命令を持つなど，８ビットとしてはかなりのもの。でも「ちょっと弱い」との声も多い。
*20**CP/M**：DOSの一種。最も有名なDOS。
*21**CP/M-68K**：68000用のCP/M。Cで記述されたことで有名。68000のDOSはUNXIになると思っていた人をくやしがらせた。が，機能的にはかなりアップしている。
*22**CP/M-09**：そんなものない。出るなんて噂すらない。
*23**FORTRAN，COBOL，LISP**：みんな言語の名前。マイコン界では嫌われている。
*24**FLEX**：DOSの一種。使い心地はまあまあ。一部ユーザーはひどく嫌っているようだが。
*25**OS-9**：UNIXのようにしたが，８ビットの限界からUNIXとし使いにくい。それでも，やっぱりすごいDOS。
*26**リエントラント**：「構造化プログラミングの嵐」によって，やたらもてはやされた手法。この例題には必ず「ハノイの塔」がでてくる。
*27**リロケータブル**：別名ポジションインディペンデント。プログラムをどのアドレスに置いても実行できるということ。

スレジスタがスタックと同様に使えて，とても便利だ。完全なリロケータブル*27にもできる。ということは，データ自体もリロケータブルにできるということ。相対ジャンプがあるうえに，プログラムカウンタリラティブなんてアドレッシングモードがあるから。16ビット転送も，ほとんど自由にできるし，演算や条件処理も強い。

とは言え，LEAX*28という理不尽な命令（と僕は思うんだけれど，真君は「いい」と言っている）があったり，レジスタ間転送が遅いとか，不満な点もある。

片やＺ80は，8080というCPUをマシン語レベルで完全に包含していて，だから8080のプログラムは全てＺ80で走らせることができ，8080のソフトウェア資産をそっくりそのまま受け継ぐことができた。そのうえで便利な命令（とくにブロック転送命令）を付け加え，相対ジャンプを可能にしたり，レジスタを増やし，とても使いやすいCPUになった。8080からＺ80にしたとき，アセンブリ言語を完全に改訂して，これが非常によかったんだね。6800から6809に変更するときも，アセンブリ言語を変えればよかったのに，とも思うね。もっとも，真君はこれには反対で，「あのアセンブリ言語はわかりやすくていい」と言ってるけれど。

ただＺ80は8080を拡張した形になっているために，相対サブルーチンコールがなかったり，アドレッシングモードがないというかあまりに貧弱だとか，できそうでできない命令が多い，フラグが変わりそうで変わらない，できそうでできないバイト単位PUSH，割り込みが強そうであまり強くない，16ビット演算が無限に弱い，とか文句はいろいろある。

「８ビットなんて，とんでもない」と言うアキちゃん*29みたいな人もいて，それを考えると，6809もＺ80もどうでもいいんだろうけれど。

ま，そんなこんなで，どちらも良いところも悪いところもあって，だから用途によって使いわけるとか，CPUにあったプログラムを書くことが，結局のところ重要なのだ。

結論――どちらのＣＰＵも良く知っていて，使い分けのできる人に，私はなりたい。

（編集部より）一部発言および注釈に，偏見があったことをおわびいたします。

---

*28**LEAX**：Load Effective Address。16ビットレジスタ演算のための命令。
*29**アキちゃん**：8086も68000も何でもペラペラ。三次元グラフィックスがとても得意。98のオーナー。本業(？)はデザイナー。

虫つくろい（第3号の訂正）

本誌第３号 114ページの「Rainy Day」のプログラムに誤りがありました。訂正し，おわびいたします（青字部分が訂正部分です）。

```
20 DIM A%(210):COLOR7,0:WIDTH80,25:DATA3,1,5,4,6,2
30 GOSUB270:FORI=0TO2000:NEXT:COLOR0:GOSUB270:COLOR7
40 BEEP1:BEEP0:FORI=0TO20STEP5:CIRCLE(319,99),I,5:NEXT
50 GET@(299,80)-(339,109),A%,G,5
```


成績処理教育評定専用プログラム《新発売》

Educational Evaluation Soft

E・E・Sと呼んで下さい！
（中学校・高等学校の先生方へお奨めします。）

※　貴方は、パーソナルコンピュータのキーボードから素点を入力することと、コンピュータがその素点に基づいてはじき出す資料から評定の段階点を決めて入力するだけで10段階（５段階）評定値一覧が得られます。
“正”の字を書きながら半日がかりで行なっていた評定作業は、昨日のこと、今日から貴方もＥ・Ｅ・Ｓの仲間になりませんか！

富士通 ＦＭ－７・８・11 ※カセットでもＦＤでもデータセーブ可	15クラス 1,000人	カセットＦＤ版 ¥5,000

データ出力内容

(1) 三種類のデータ分布表（ヒストグラム）
※認意の級の幅によるヒストグラム
※10段階相対評価にもとづくヒストグラム
※標準偏差にもとづく10段階のヒストグラム

(2) 処理結果の出力
※評定値一覧表
クラス番号・出席番号・素点・偏差値・10段階・５段階・評定値・順位
※クラス別評定値分布表
学期末・学年末（学期末のデータ加算）各々成績評定が行えます。

◎実際のPRINTONT時「B:シュッセキバンゴウ」の後に名列表がピッタリ当てはまる様スペースがあいて出力されます。

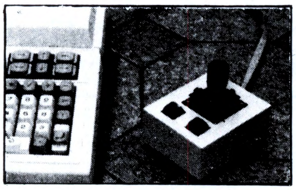
ウーン これは便利だ！

ＦＭ－７用　¥3,500
パチスロマシンゲーム
君の７がスロットマシンに早がわり！ボーナスゲームでさらに楽しめるサウンド付

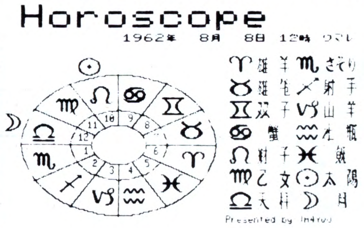
ＦＭ－８・７　¥3,500
グラフィックビリヤード
本格的グラフィックの四ッ球ビリヤード
球の打つ位置・方向・強さの設定可

待望の
ＦＭ－７用ジョイスティック新発売
テンキー対応（２・４・６・８と１・３・５の切換式）
スーペスキーとストップキーはPUSHボタン式
※オートリピートスイッチ付
（ギャラクシャン等のソフトでジョイスティックと画面の動きとが連動します。）　¥12,800

ＦＭ－８ テンキー対応 限定発売
ジョイスティック　¥12,800
お早目に！仕様はＦＭ－７用と同じです。



“だまってすわればピタリと当たる”
ＦＭ－７・８・11　DISK版
「パソコン星占術」　¥９,８００
※生まれた年・月・日・時間から太陽・月・星座の位置を計算し完璧なホロスコープを作成し性格・相性・適性など全て診断。プリンター用紙約３枚分の判定出力を致します。



UE URBAN
〈開発・製造元〉
アーバン電子株式会社
〒730 広島市中区大手町1－1－23　☎(082)246-0993㈹

はじめて握るはんだごて

インタフェースを作ろう

第1回

志沢雅彦

キーボードからの入力，CRTへの出力，これだけでは何んとも閉鎖的なマイコン使用法です。FMにもっといろいろな入出力装置を付けて，いろいろなことをさせてみませんか。

第1回は，そのためのインタフェース作りです。第2回以降では，このインタフェースを使って，FMにジョイスティックを付けたり，タイマをコントロールする予定です。

1 はじめに

この記事は，次のような人を対象にしています。

① 初心者向けです

はじめてはんだごてを使って電子工作をやってみようと思っている人が，楽に入門できるようになっています。もちろん，今までに電子工作を多少ともやったことのある人ならば，容易に組み立てることができます。ただし，ハードウェアの自作は，ソフトウェアと違って，一歩誤ると愛機を壊してしまうことにもなりかねません。あわてず，慎重に作りましょう。

② FM-7,8とも共通に使えます。

ここで製作するハードウェアは，FM-7，FM-8の後部にある拡張バス端子（expansion port，図1参照）に接続します。この端子はFM-7,8とで多少異なる部分もありますが，この記事では，気にすることなくどちらでも使えるようになっています。

また，この拡張バス端子を使用して，フロッピディスクを使っているFM-8ユーザーや，上部スロットではなく拡張バス端子にフロッピディスクを接続しているFM-7ユーザーでも，フロッピディスクと共用できる方法を示します。

2 インタフェースとは

そもそもインタフェースとは何んでしょう。**図2**を見てください。これは，マイクロコンピュータの基本的な内部構成図です。CPU（FM-7，8では6809が使われています）からバスと呼ばれる信号線の束が出ています。CPUはこのバスを通して，キーボード，ディスプレイ，カセット，ディスクドライブなどの入出力装置とデータのやりとりをします。また，バスはROM，RAMなどのメモリとの間のデータのやりとりも行います。しかし，バスに直接，キーボードやディスプレイなどを接続するわけにはいきません。なぜなら，バスの信号の形式と，キーボード，ディスプレイなどの信号の形式が全く異なるからです。そこで，バスとこれらの入出力装置間の仲介役を務める回路が必要となります。それがインタフェースと呼ばれる回路です。

例えば，キーボードそれ自体は，キースイッチが集まっただけのものです。そこでキーボードインタフェースは，それぞれのキースイッチに電流が流れているかどうかを常に調べていて，キースイッチが押されると，CPUにバスを通してそのことを知

図1

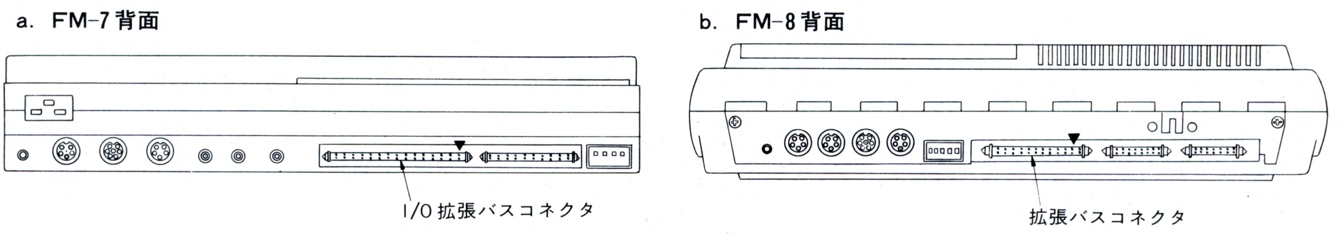
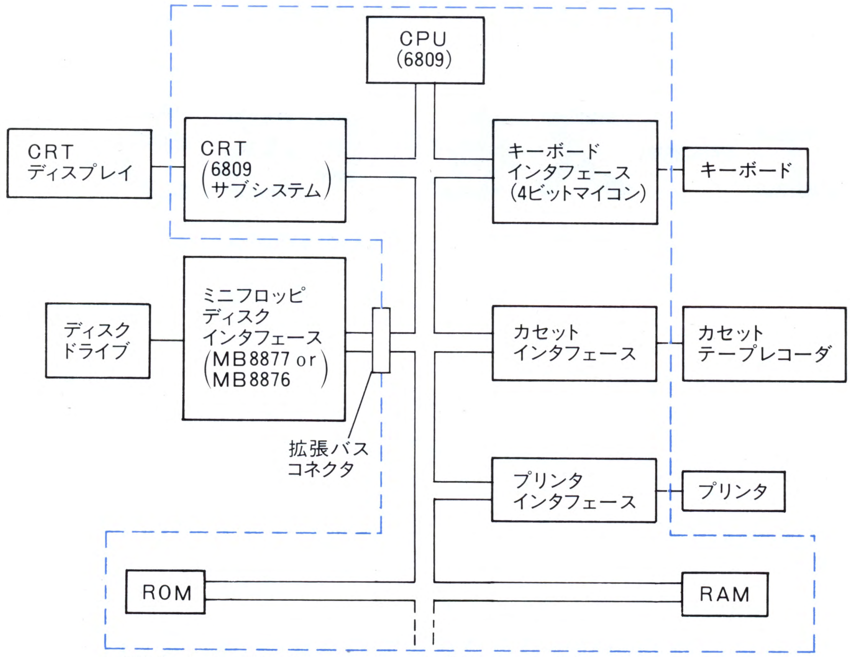


図2　マイクロコンピュータの内部構成



＊説明のために簡素化してありますが，実際はもっと複雑です。
＊ ⬚ はFM-7,8本体内にある部分

表1　6809ファミリLSI

6821	PIA	（ペリフェラルインタフェース・アダプタ）
6844	DMAC	（ダイレクトメモリアクセス・コントローラ）
6845	CRTC	（CRTコントローラ）
6850	ACIA	（非同期通信インタフェースアダプタ）

写真1　本体と接続したところ

らせ，どのキースイッチが押されたかという情報を，バスを通してCPUに送る役目をしています。ちなみに，FM-7，8では，キーボードインタフェースに4ビットのワンチップマイコン*が使われています。

ところでFM-7，8はCPUに8ビットCPUである6809を使っています。このほかに8ビットCPUには，Z80，8085などがありますが，これらのCPUのバスはそれぞれ違った規格になっています。また，バスと入出力装置との間のインタフェースは，普通，かなり複雑な回路になります。そこでよく使うインタフェース回路はLSIになっていて，6809，Z80，8085などそれぞれのCPUは，“ファミリLSI”というそれぞれのバスにマッチしたファミリをもっています。

表1に，6809の代表的なファミリLSIを示します。「インタフェースを作ろう」では，6821という6809用ファミリLSIの高速版である68B21を使って，自前のインタフェースを作ります。この68B21は，PIA（Peripheral Interface Adapter）と呼ばれているもので，デジタル信号のオン・オフの情報を，バスと入出力装置の間で，そのままの形でやりとりします。今回はこのLSIを使って，LED**を点滅させて，文字を表示させてみましょう。68B21の詳しい説明は次回行います。今回は，まず作ってみてください。

3 インタフェースを作る

今回製作するものは，**写真1**のように，PIAを乗せた基板と，LEDに電流を供給するための回路をのせた基板，そして，LEDを取り付けた棒です。この棒を左右に振ると，目の残像効果で文字が見えるわけです。

今後，このPIA基板を共通に使って，いろいろなものをつなげていきます。

繰り返しますが，ハードの製作はちょっとしたミスが素子を壊すことになりますから，本文をよく読んでから作り始めてください。

◆工具をそろえる◆

まず工具類をそろえます。次のものは最低限必要です（**写真2**参照）。

①ニッパおよびワイヤストリッパ
②はんだごて，はんだごて台
③ピンセット
④はんだ（なるべく細いものがよい）
⑤カッタ

また，テスタ（**写真3**）は今後ハードを作っていくためにも，また理解するためにもたいへん役に立ちますから，まだ持っていない人はこの機会に1台買うことをおすすめします。アナログ式ならば2,000～6,000円くらい，デジタル式でも最近は8,000円ぐらいから手に入ります。次の部品集めと一緒にやってください。

＊一つの石に，CPUとROM，RAMなどのメモリが組み込まれたものをワンチップマイコンという。これに対し，1枚の基板にCPUと，ROM，RAMなどをのせたものを，ワンボードマイコンという。
＊＊発光ダイオード。FMシリーズの電源やCAPなどの状態を表示する赤いランプ状のものもLED。色は赤・黄・緑などがある。

◀写真2　工具類

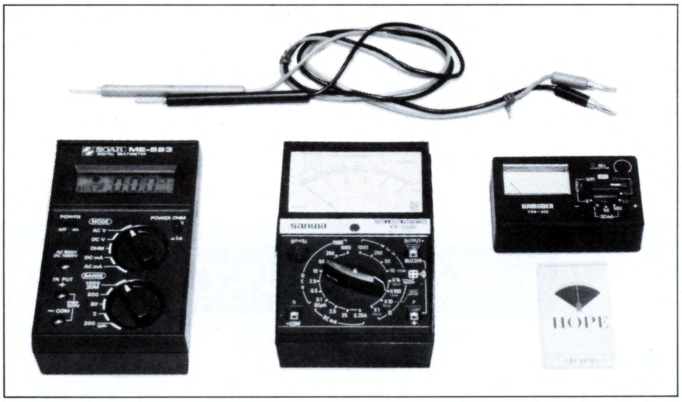
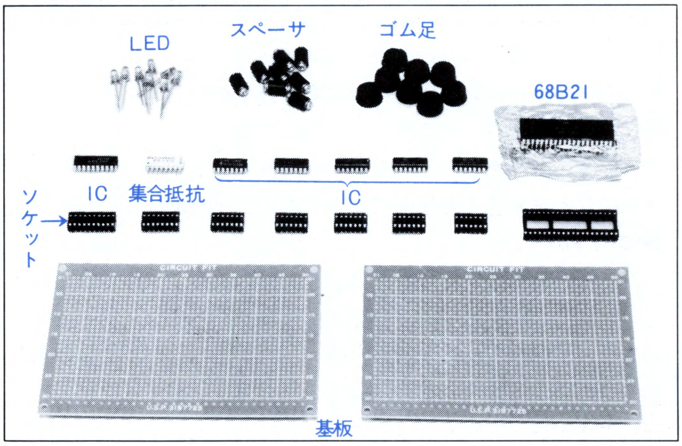


写真3　テスタ▲

▼写真4　部品(このほかにコンデンサ，ヘッダ，フラットケーブルがある)



◆部品を集めよう◆

部品リストを**表2**に示します。部品集めは，東京近辺の人は秋葉原，大阪近辺の人は日本橋に買いに行きましょう。その際，本誌を店員さんに見せれば，親切に教えてくれる所も多いと思います。今回，私は亜土電子でそろえましたが，もちろんほかのお店でも集められます。また，地方の人は通信販売で買うのがいいでしょう。「トランジスタ技術」などのパーツ屋さんの販売広告がたくさん載っている雑誌を活用してください。

◆さあ，作ろう◆

図3のフローチャートにしたがって，1ステップずつ確認しながら進めましょう。はじめに**PIA基板を製作**します。

① ICソケットの取り付け

ICは基板に直(じか)付けすると，熱などでICを壊すことがあります。ICソケットをはんだ付けして，それにICをはめ込むようにします。

まず，**図4**の部品配置図を見ながら，ICソケットをセロハンテープで基板の表（銅はくの付いていない面）から張り付けて仮止めします（**写真5**）。このとき，ソケットのピンがすべて裏まで突き出ているかどうか確認しながら行ってください。ソケットの向きにも気を付けてください。

次に，ICソケットをはんだ付けします（**写真6**）。このとき，端から順々に行わないで，まず対角線の位置にある二つの端子をはんだ付けし，ICソケットが基板から浮いていないかどうか確認してから残りのピンをはんだ付けします。はんだを多く使いすぎて，ショートさせないように気を付けてください。

② ヘッダの取り付け

ヘッダもICと同様な手順で基板にはんだ付けします。ディスクインタフェースを取り付ける人は，

表2　部品リスト

品　　名	型　番（メーカー）	個　数	販売店	単　価
IC	68B21（モトローラ・日立・富士通ほか）*1	1	亜土電子ほか	680
	74LS04　（各社）	1	〃	40
	74LS30　（〃）	1	〃	35
	74LS138（〃）	1	〃	75
	74LS245（〃）	1	〃	210
	7416　（〃）	2	〃	70
ICソケット	14P（バーンディ・JAEほか）	4	〃	45
	16P（〃）	2	〃	50
	20P（〃）	1	〃	65
	40P（〃）	1	〃	100
LED	TLUR153(赤)，TLUG153(緑) TLUY153(黄）いずれか（東芝)*2	8	〃	70
集合抵抗	898-3-R150（ベックマン）	1	〃	200
電解コンデンサ	100μF 16V（各社）	1	〃	100
コンデンサ	0.01μF（セラミック）	6	〃	10
コネクタ	フラットケーブル付 50P*2	1組*3	*2	1580
	〃　26P*2	1組	*2	980
ヘッダ	50P*2	1 *2,3	*2	480
	26P*2	2 *2	*2	320*4
基板	ADO725B（亜土電子)*2	2	亜土電子	250*2
スペーサ	1cm，ネジ付き	8	亜土電子ほか	30
ゴム足	径15mm程度のもの	8	〃	10
配線材料	ラッピングワイヤ	*2	〃	*2
	フラットケーブル 10本	3m	〃	200/m*5

価格は，お店によって多少異なります。一応の目安にしてください。
なお，亜土電子では，Oh! FMのPIA製作記事の部品キットと言えば，PIA基板用5,200円，同ディスク兼用用7,400円，LEDドライバ基板・LED基板用2,400円で販売してくれます。（ただし送料は350円ぐらい）

*1 モトローラはMC68B21PまたはMC68B21L，日立はHD68B21PまたはHD68B21ですが，富士通の型番はMB8874Hです。
*2 「部品について」を参照してください。
*3 ディスクなど拡張バスをほかに使う人は2組。
*4 FAP26-03＃2
*5 スダレの場合 250円/m

ディスク側のヘッダは，まだ取り付けないでください。

③　コンデンサの取り付け

コンデンサC_1〜C_6を取り付けます。電解コンデンサは⊕，⊖があります。**図5**に⊕，⊖の見方を示しておきます。コンデンサのリード線は先に切らずに，まずそのまま基板の穴に差し込んで，はんだで基板に固定してから，ニッパで不要な部分を切り取ります。

④　配線

ここが最も時間のかかるステップです。**図6**の実体図を見ながら，1本ずつ確実にくっついているかどうか軽く引っぱってみながらすすめましょう。はんだは多ければいいというものではありません。多すぎるとかえってすぽっと抜けてしまうことがあります。また，ラッピング用ワイヤは，は

部品について

●IC 68B21

モトローラ，日立のものが手に入りやすいようです。富士通のものは型番がMB8874Hです。パッケージはプラスチックのもので十分です。このLSIは，NMOS形のものなので静電気によって破壊されることがあります。お店ではアルミはくに包むか，導電マットに差した形で渡されますので，ソケットに取り付けるまでは，そのままで保存しておいてください。FM-8で倍速の改造を行っていない人は68A21でもかまいませんが，将来のことを考えて68B21で作っておくことをおすすめします。

●IC TTL

74LS××あるいは74××のICはTTLと呼ばれているものです。このICは各社から出ていますが，どのメーカーのものでもかまいません。ただしメーカーによってICのピン番号を示すマークが異なるので，ソケットに差し込むときに注意してください。

●ICソケット

ICソケットは，高級なものから安価なものまで各社から各種出ていますが，テスタの棒をあてやすいものでは，バーンディとJAEのものがあります。バーンディの方が安価ですが，JAEのものはがっちりした感じです。このどちらかをおすすめします。

●LED

LEDは普通に売っているものなら，だいたいどれでも使えますが，ピンの意味がまちまちなので，ここでは東芝のものにしてください。今回は，高輝度発光ダイオードのウルトラブライトシリーズを使いました。

●集合抵抗

配線を楽にするため，ちょっとぜいたくですが集合抵抗を使いました。これは150Ωの抵抗が８本入ったものです。最近では，ベックマンのほかにも同種のものが出回っていますので，それでも結構です。

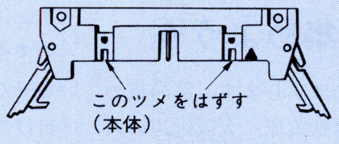
●電解コンデンサ

100μF 16Vの電解コンデンサには，リード線が同じ方向に出ている縦型と，反対方向に出ている横型があります。私は横型を使いましたが，縦型の方が手に入りやすいので，それでも構いません。

●コネクタ＋フラットケーブル

FM-7，8本体とPIA基板をつなぐ50芯フラットケーブルは約40cmぐらいのものを使います。お店でコネクタを両端に圧着してもらってください。FM-7，8本体側のコネクタは富士通製か3Mのものにしてください。どちらもない場合には，本体側のヘッダについている保護用の白いツメを取りはずす必要があります（**図A**参照）。ただし，以後使用するときには，方向に注意してください。

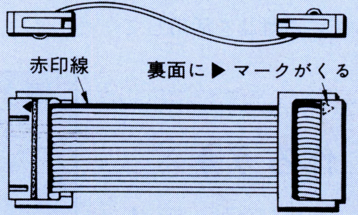
図A



本体側のヘッダにつなぐコネクタ以外のコネクタは，どのメーカーのものでもかまいませんが，接続するヘッダと同じメーカーのものにしてください。

コネクタをフラットケーブルに圧着してもらうときの方向は，**図B**のようにしてもらってください。

図B



PIA基板とLEDドライバ基板をつなぐ26芯のフラットケーブルは15〜30cmぐらいで，両端に50芯のものと同じ方向でコネクタを圧着してもらいます。

●ヘッダ

フラットケーブルに圧着してもらったコネクタと同一メーカーのものを買ってください。

●基板

私は，亜土電子製のADO725Bという140×90mmの大きさのユニバーサル基板を使いました。同程度の大きさであれば，どのメーカーのものでも構いません。サンハヤトの基板が手に入りやすいでしょう。

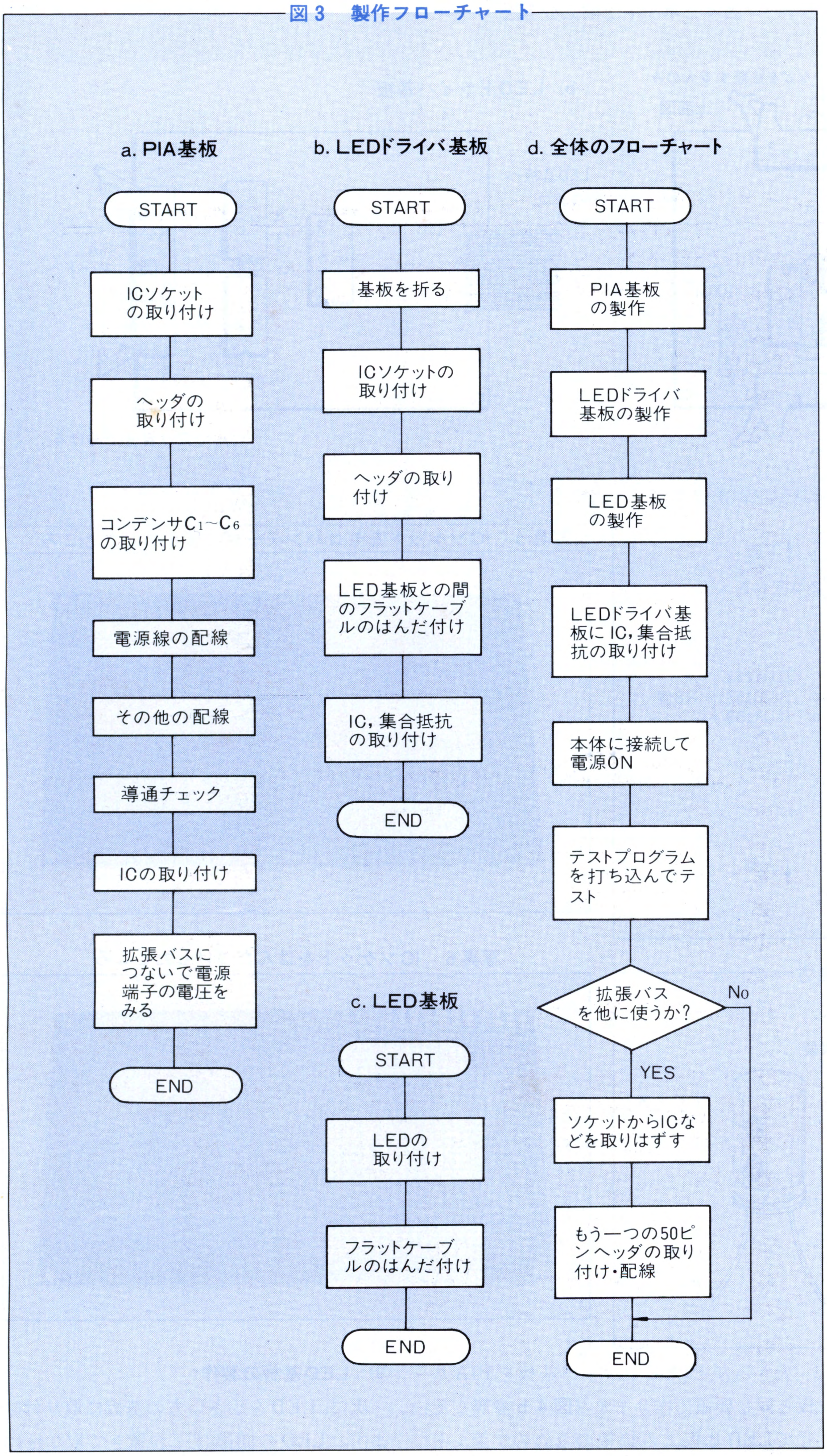
●スペーサ，ゴム足

これは基板の足に使うものです。

●配線材料

ラッピング用のワイヤは，0.3mmのテフロンワイヤを使いました。これからの連載のために，ボビンに巻かれた100m巻のものを買っておいてください。1,500円ぐらいです。

図3　製作フローチャート



んだ付けの前に切ってしまわずに，まず一方を取り付けてから，必要な長さに切って，もう一方をはんだ付けするようにしてください。このとき，これからはんだ付けする場所の上をリード線が通らないように，特に気を付けてください。

配線の順序は，実体図の基板に近い方，つまり下の方から行ってください。普通，はじめに電源線を配線します。あとは，まず距離の短いものから順々にやっていけば，きれいにまとまります。また必ずしも，**図6**の実体図のとおりに配線する必要はなく，基板の表側を使ってもかまいません。**図7**に回路図を示します。

⑤　導通チェック

まず，それぞれのIC，ヘッダのピンについて，配線されているピンと配線されていないピンが実体図と一致しているかどうかを見ます。

次に，それぞれのICの電源ピンの間がショートしていないかどうか調べます。このとき，テスタは抵抗測定の最低レンジにします。電源間にコンデンサが入っているため，若干，針が振れますが，0Ωでなければ大丈夫です。針が振り切れるようでしたら，C_2などのピン間を一度ショートしてからやってみてください。また，ヘッダの電源ピンとの間の導通も調べます。

⑥　ICの取り付け

次はICの取り付けですが，ICの向きには十分に注意してください（**図8**参照）。これを逆にすると，まず必ずそのICを壊してしまいます。**図4**の部品配置図を見ながら，ソケットにはめ込んでください。

⑦　拡張バスに接続してテスト

いよいよ電源を通してみます。まず，FM-7，8本体の電源が切ってあることを確かめてから，50ピンのコネクタがついたフラットケーブルを使って，本体の拡張バス（**図1**参照）と，PIA基板をつなぎます。このときICの方向と同様，コネクタの方向に十分注意してください。▲印と▲印が合うように，また赤（青のケーブルもある）の線が必ず1番ピンにくるようにします。

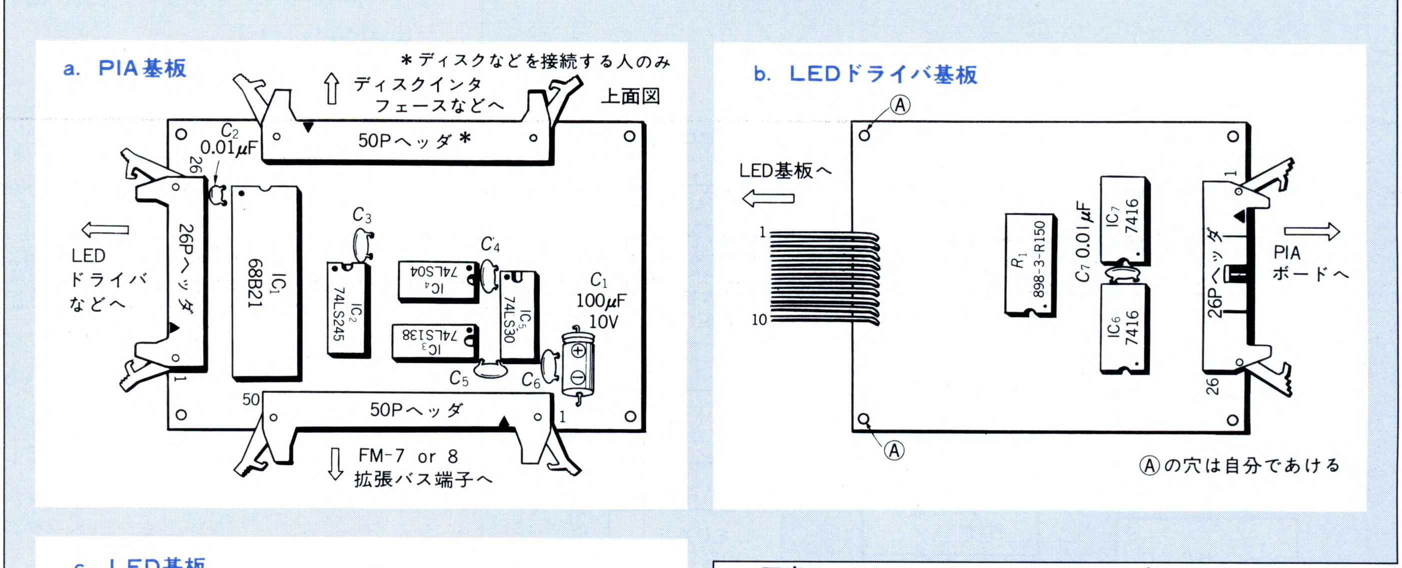
ディスクなど，拡張バスに接続するものを持っている人も，この段階ではつなぎません。

まずROM－BASICモードにディップスイッチをセット（FM-7は1，2をON，FM-8は9，10をON）してから，電源を入れます。もしこのときBASICのメッセージが出なかったり，異常がある場合は，直ちに電源を切って配線，ICの向き，コネクタの接続の向きを調べてください。

次に，テスタを使って各ICの電源端子間（V_{cc}とGNDまたはV_{ss}の間）の電圧を測定し，＋5Vがかかっているかどうかを調べます。

ここまで終えたら，電源を切ってLED基板ができあがるまでそのままにしておきます。

図4　ボード上部品配置図



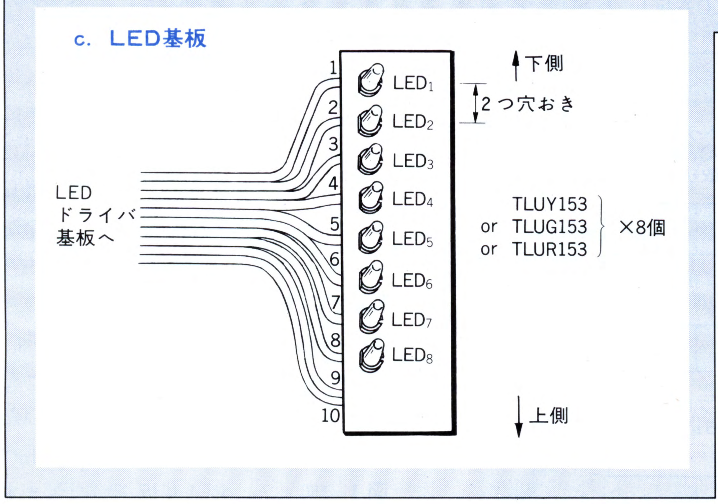


写真5　ICソケットをセロハンテープで仮止めしたところ

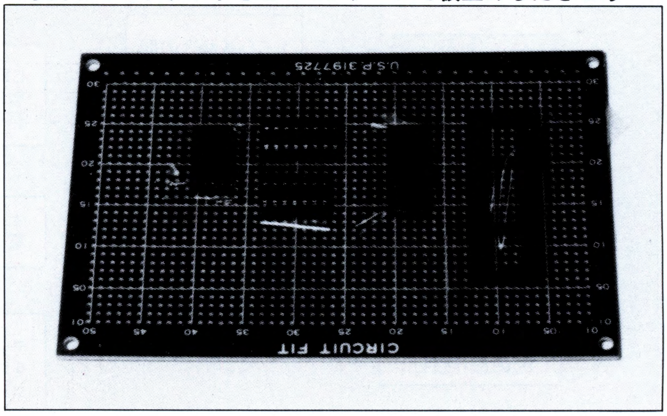


写真6　ICソケットをはんだづけしたところ

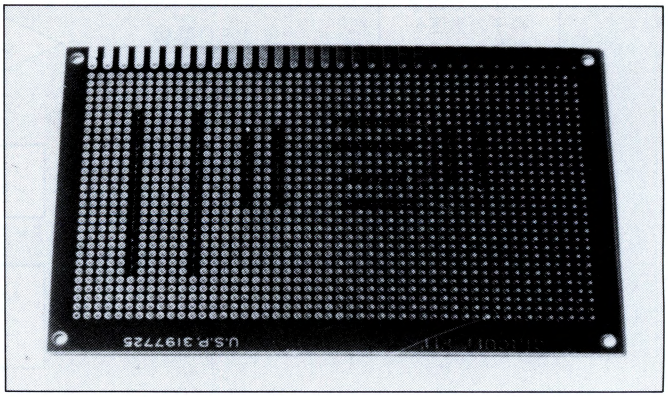
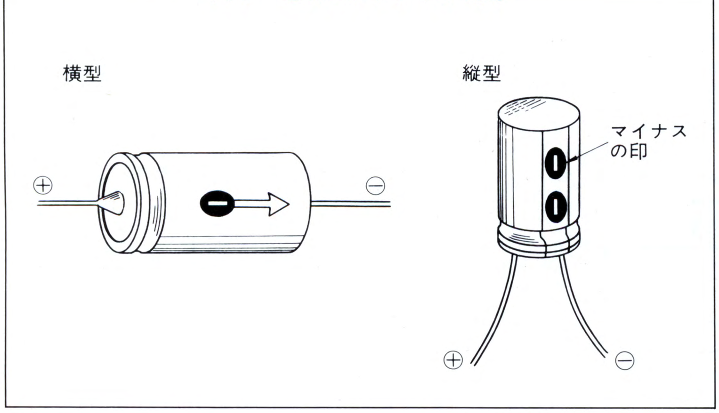


図5　電解コンデンサの見方



⑧　LEDドライバ基板の製作

LEDドライバ基板とLED基板は，1枚の基板を二つに折って使います（**写真7**）。まず，カッタを使って基板の表裏に傷を付けます。このとき定規をあてると楽にできます。そして，机の端に切り離す線がちょうどくるようにしてあてがい，ちょっと強めに曲げると簡単に折り切ることができます。

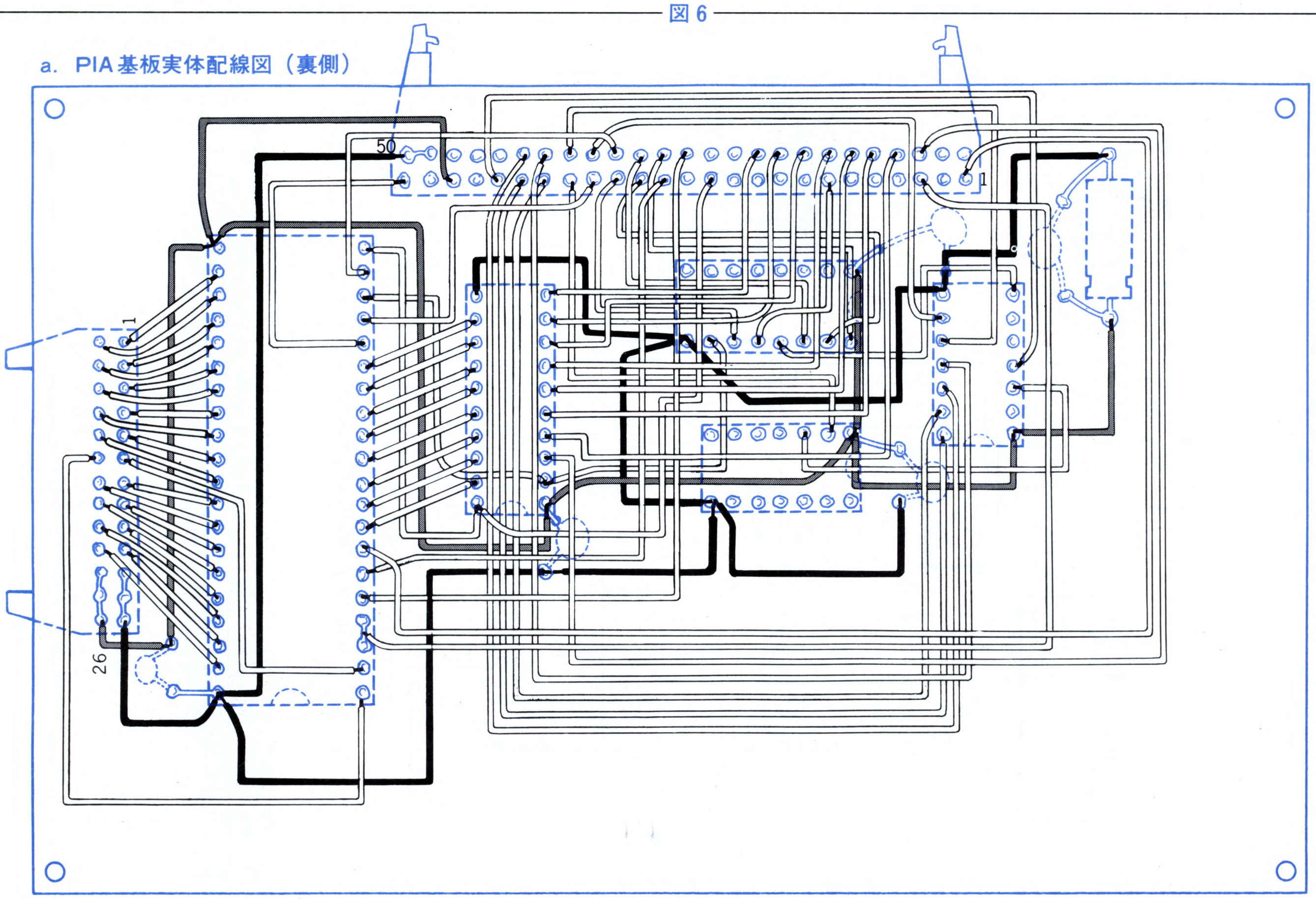
大きい方でLEDドライバ基板をPIA基板と同じ要領で作ります（**図4 b**参照）。そしてLED基板との接続のためのフラットケーブルを基板にはんだ付けします。このとき，10本すべてを互いにほぐして被覆をむいてしまってからやってください。そして，このフラットケーブルと集合抵抗がのるソケットとの間を配線します（**写真8**）。

⑨　LED基板の製作

次に，LEDを小さい方の基板に取り付けます。LEDの間隔は二つ置きで取り付けてください。まずLEDを差し込んで足を折り曲げます。このとき，アノード（**図9**参照）は共通なので互いに重なり合うように曲げ，カソード（足の短い方）は基板の外側に向けて曲げます（**写真9**）。そしてLEDが基板に対して直立するように気を

図 6

a. PIA 基板実体配線図（裏側）



b. LED ドライバ基板配線図（裏側）

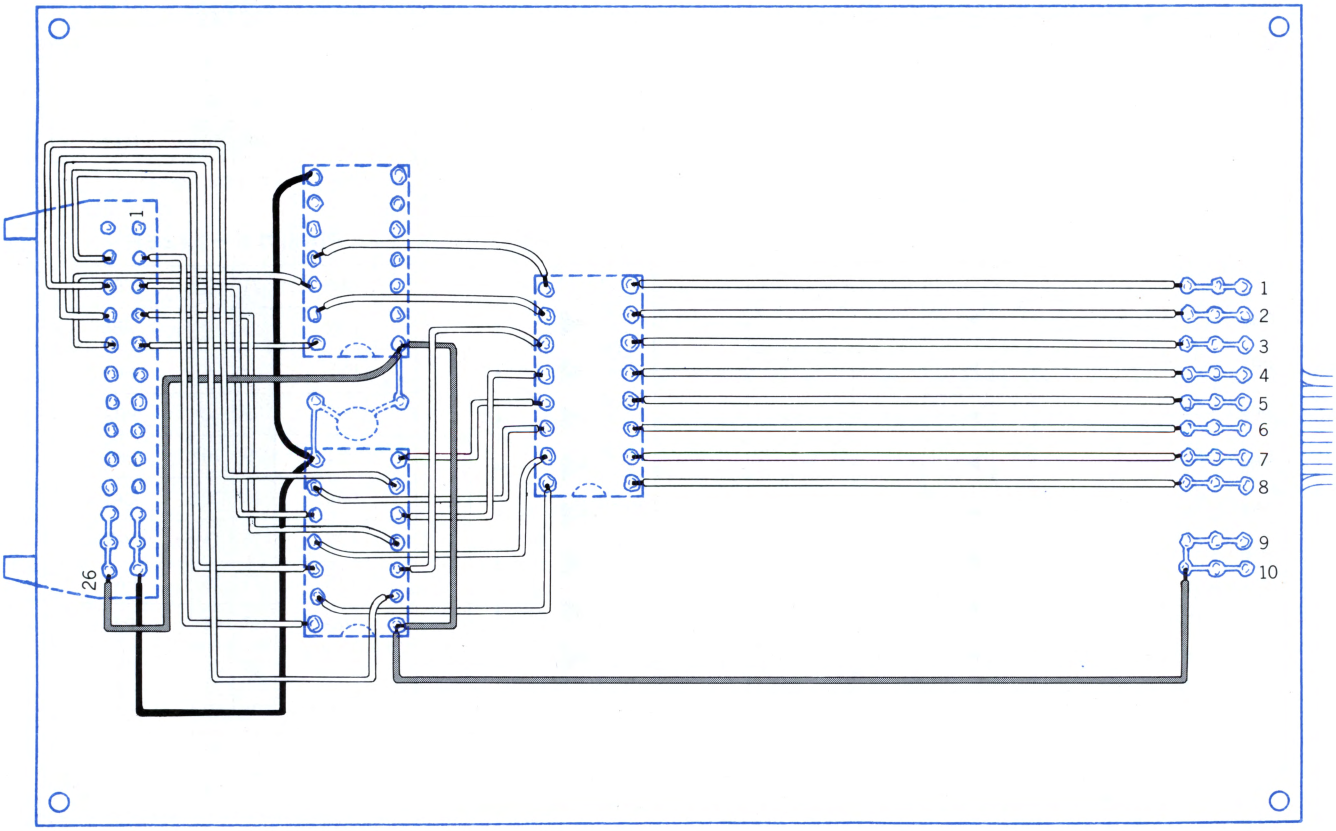
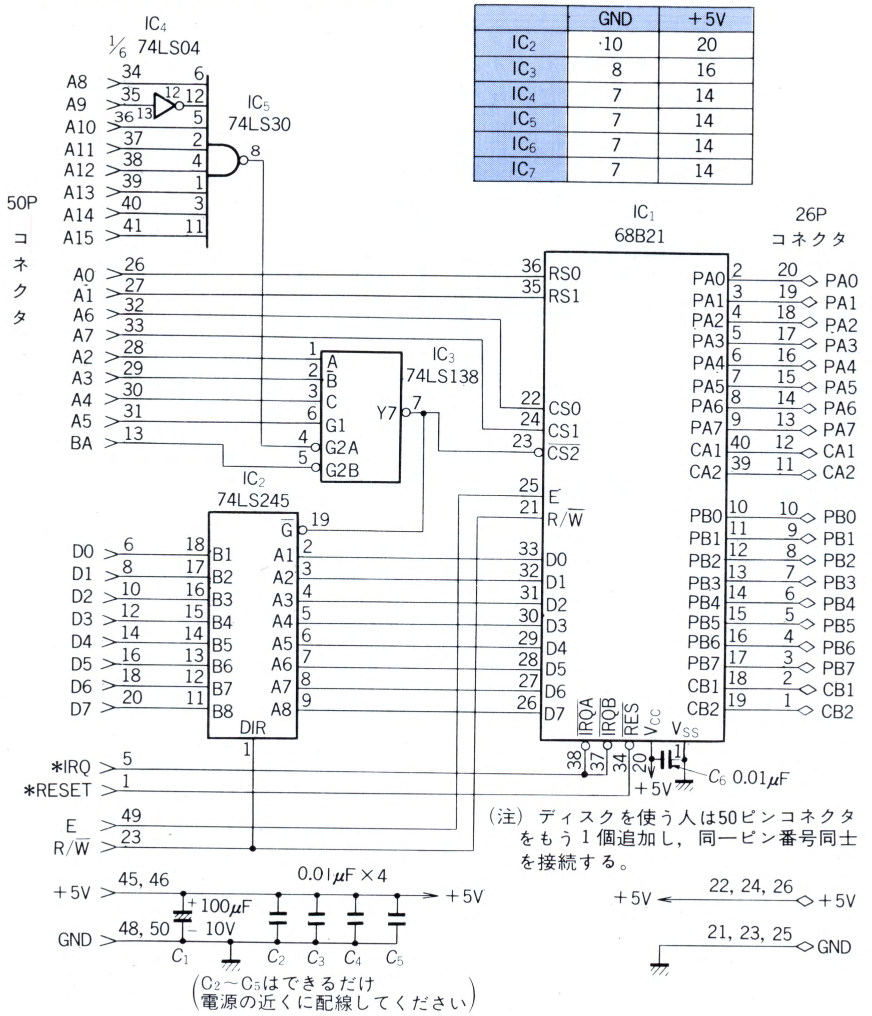


図 7

a. パラレルインタフェース・ボード

電源ピンの接続

	GND	+5V
IC2	10	20
IC3	8	16
IC4	7	14
IC5	7	14
IC6	7	14
IC7	7	14



b. I/O拡張ポート(50P)

*RESET	1	2	(*RAMIHB)
	3	4	*IOSEL
*IRQ	5	6	D0
*NMI	7	8	D1
*FIRQ	9	10	D2
(GH)	11	12	D3
BA	13	14	D4
BS	15	16	D5
(*REFREQ)	17	18	D6
(REFGRNT)	19	20	D7
(REFCK)	21	22	*EXTDET
R/W	23	24	GND
GND	25	26	A0
A1	27	28	A2
A3	29	30	A4
A5	31	32	A6
A7	33	34	A8
A9	35	36	A10
A11	37	38	A12
A13	39	40	A14
A15	41	42	(TASK0)
*MRDY	43	44	
+5V	45	46	+5V
Q	47	48	GND
E	49	50	GND

(　　)内はFM-8のみ

PIA出力ポート(26P)

CB2	1	2	CB1
PB7	3	4	PB6
PB5	5	6	PB4
PB3	7	8	PB2
PB1	9	10	PB0
CA2	11	12	CA1
PA7	13	14	PA6
PA5	15	16	PA4
PA3	17	18	PA2
PA1	19	20	PA0
GND	21	22	+5V
GND	23	24	+5V
GND	25	26	+5V

c. LEDドライバボード，LEDユニット

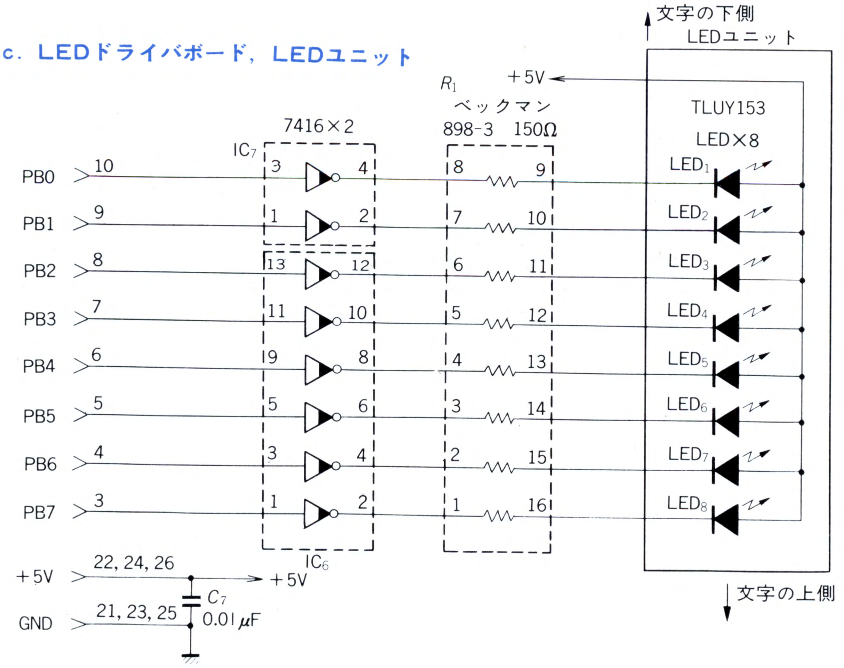


図8　各ICのピンの接続図（上から見た図です）

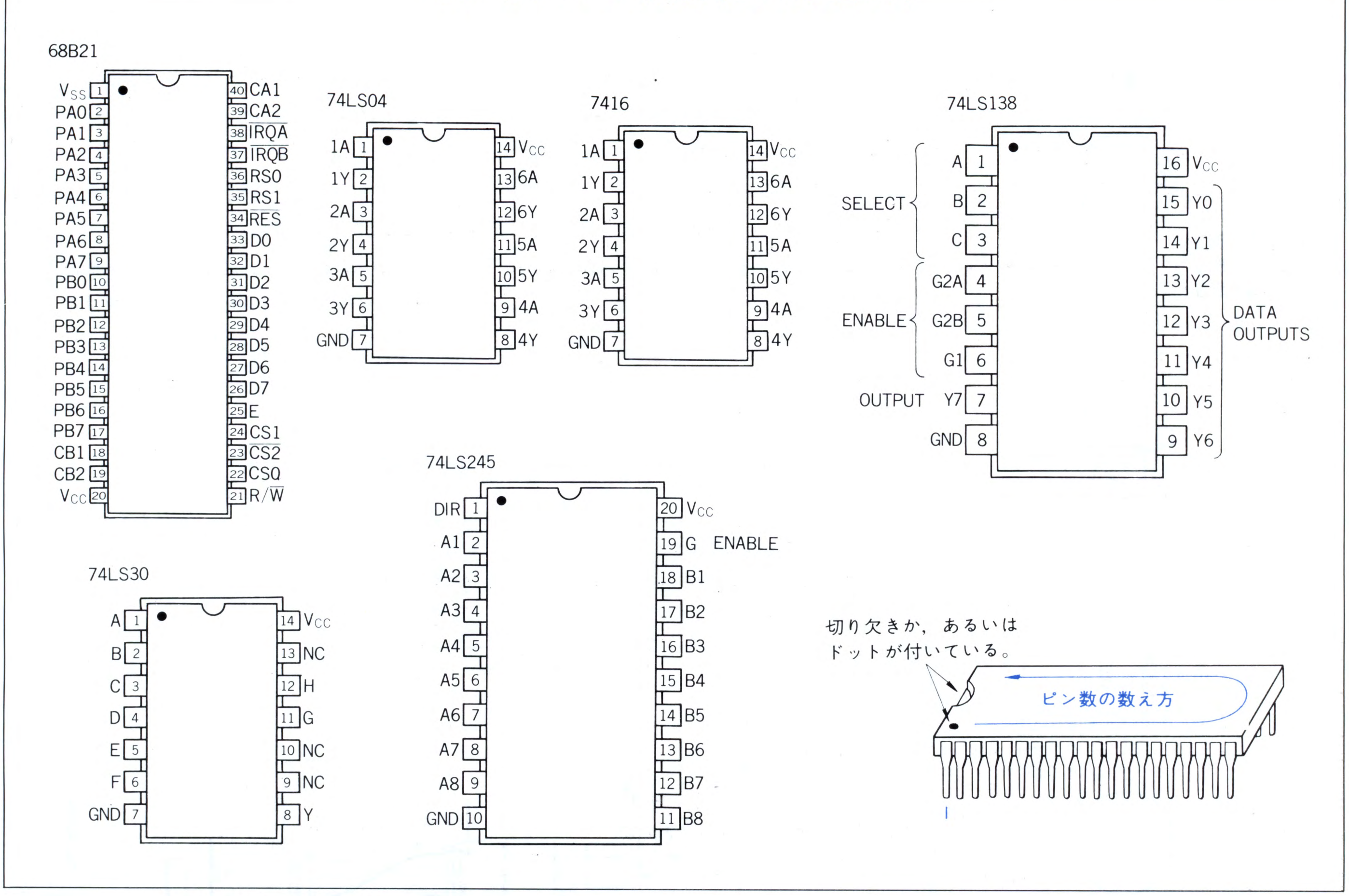
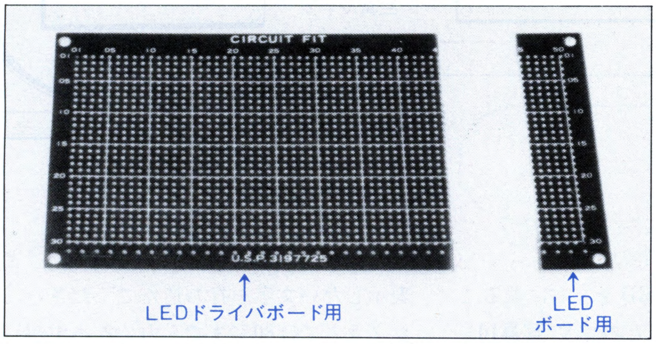


写真7　基板を折ったところ



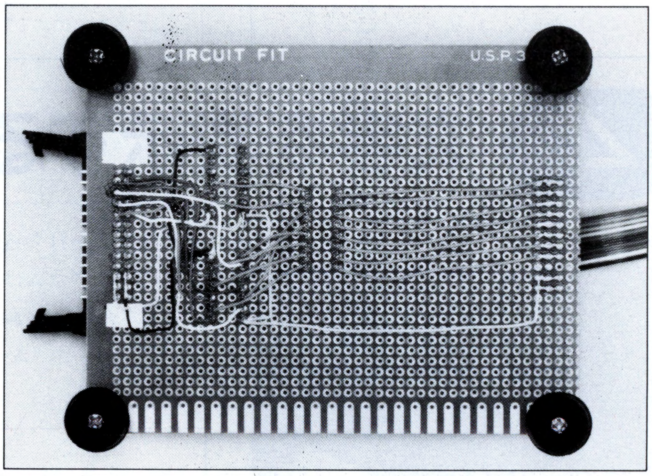
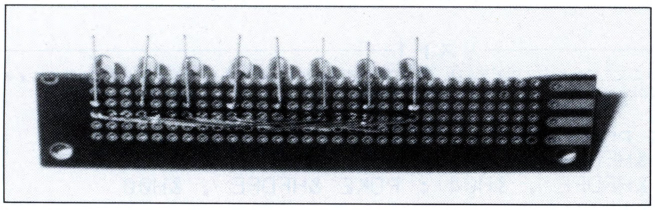


写真9　LEDのとりつけ



付けて，はんだ付けをしてください。カソードのリード線のうち，基板からはみ出した部分は，ニッパなどでカットしておきます。そして，LEDドライバ基板からのフラットケーブルをはんだ付けします。

⑩　LEDドライバ基板にIC，集合抵抗を取り付ける

PIA基板と同じようにIC（7416）と集合抵抗を向きに注意して差し込みます。

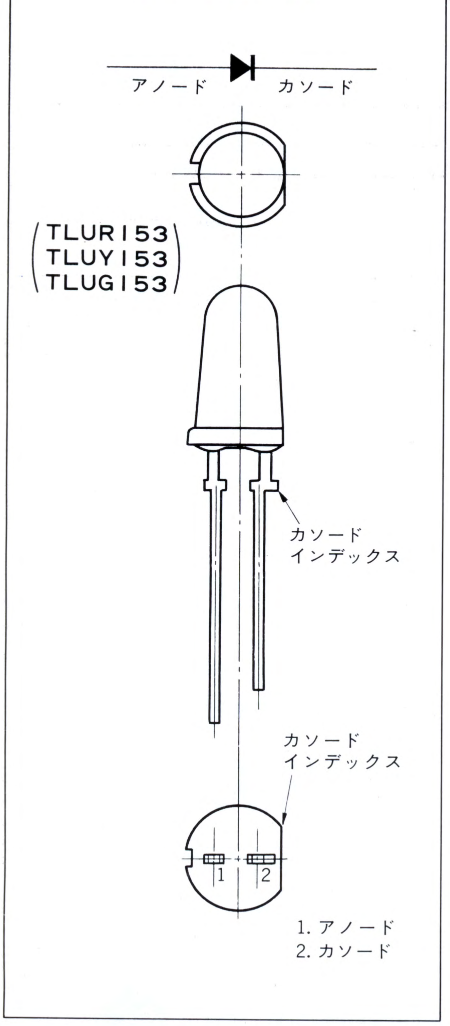
⑪　LED基板をPIA基板に接続してテストする

26ピンのコネクタのついたフラットケーブルを使って，LEDドライバ基板とPIA基板を接続してください。ここでもコネクタの方向には注意しましょう（**図10**参照）。

次に，FM-7，8本体の電源を入れてみてください。全てのLEDが点灯するはずです。

次に**リスト1**を打ち込んでRUNさせてください。LEDが下から上へ流れるように一つずつ点滅すれば成功です。もし，全く点灯しなかったり，点灯の順番・方向が

図9　LEDの見方



おかしいときは，直ちに電源を切って配線のチェックを行います。

⑫　**完成！**

あとはLED基板を30〜50cmほどの棒に取り付けます。**写真1**のものは，ICを入れるケースを使いましたが，どんな棒でもかまいません。このとき，方向を間違えると，あとで文字が逆さまに出てしまいますので気を付けてください。

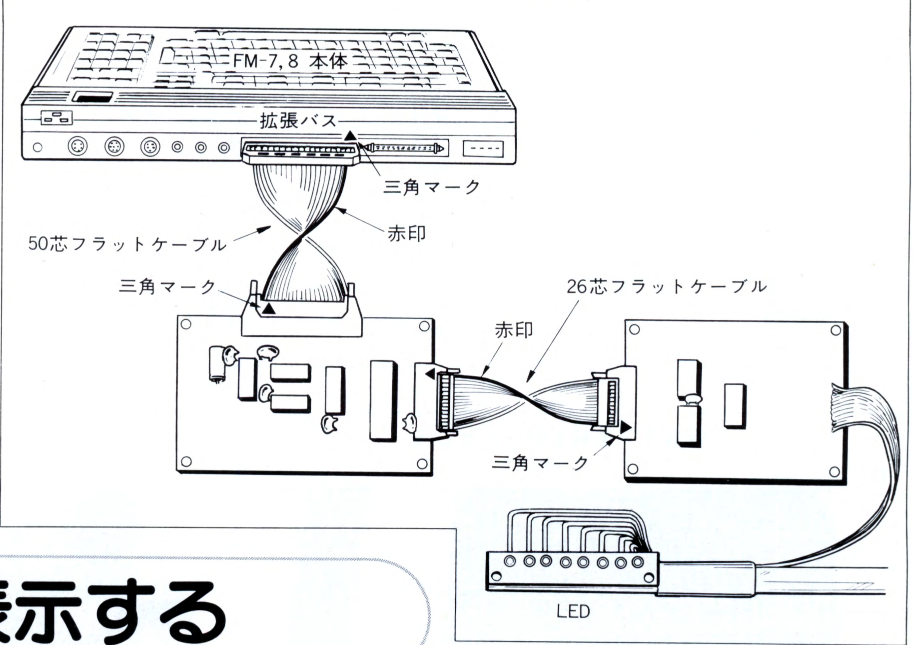
PIA基板，LEDドライバ基板それぞれに，ゴム足とスペーサを取り付けてください。

拡張バスに，PIA基板以外接続するもの(ディスクなど)がなければこれで完成です。

拡張バスにディスクなどをつなげる人は，まずPIA基板のICをソケットから抜き出します。抜かずにすると，古いはんだごてなどでは，漏れ電流によってICを破壊することがあります。

さらに50ピンヘッダをPIA基板に取り付け，もう一方の50ピンヘッダのピンとつなげます。1ピンは1ピン……というように対応したピン同士を配線します。

図10　各ケーブル接続図



4 文字を表示する

リスト2を打ち込んでください。

このプログラムは，LEDを左右に振ることにより文字を表示するものです(**写真10**)。RUNさせると，表示する文字を聞いてきますから，Oh！FM⏎とか，ABC⏎とか表示したい文字を打ち込んでください。プログラムでは80字まで入力できますが，腕を振る距離を考えれば5文字ぐらいまでが

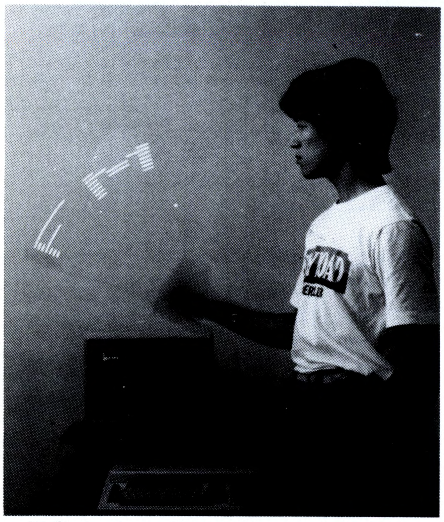


写真10　LED基板を振って文字を表示

リスト1

○ここで扱われるプログラムは，個人で利用するほかは著作権法上 無断複製を禁じられています。
COPY RIGHT © 1983 M.SIZAWA

```
10 'test program for LEDs
20 POKE &HFDFF , &H00 : POKE &HFDFE , &HFF
30 POKE &HFDFF , &H04 : POKE &HFDFE , &H00
35 WHILE 1
40 I = 1
50 WHILE I < &HFF
60  POKE &HFDFE , I
70  I = I + I
80  FOR II = 0 TO 200 : NEXT II
90 WEND
100 WEND
```

適当でしょう。Yまたはyで実行しはじめます。ピッという音に合わせて左右に棒を振ると文字が見えます。はじめのうちはなかなか調子が合わないでしょうが，何度か練習してみてください。**リスト2**のFOR～NEXTのループ回数を変更して，自分に合ったタイミングに変えてもよいでしょう。

5 さいごに

次回は，今回使ったPIAの使い方の説明と，ジョイスティックの製作です。お楽しみに。

今回製作したPIA基板などの部品を5名の方にプレゼントいたします。詳しくは，113ページをご覧ください。

リスト2

```
100 ´
110 ´************* LED display application (2) *******************
120 ´
130 ´     ver. 1.0  1983/5/20            by M.Shizawa
140 ´
1000 DEFINT A - Z
1010 WIDTH 80 , 25
1020 ´initialize PIA port B
1030 POKE &HFDFF , &H00 : POKE &HFDFE , &HFF ´set all bits output
1040 POKE &HFDFF , &H04 : POKE &HFDFE , &H00 ´set all LEDs off
1050 ´input string
1060 PRINT "LED display application" : PRINT
1070 A$ = "" : WHILE A$ = ""
1080 LINEINPUT " string for display ? " ; A$
1090 WEND
1100 CLS
1110 LOCATE 0 , 0 : PRINT A$ : PRINT ´display string
1120 DIM PATTERN( LEN( A$ ) - 1 , 8 ), TEMP( 0 ) ´reserve pattern array
1130 ´get character pattern
1140 FOR X = 0 TO LEN( A$ ) - 1
1150   FOR XX = 0 TO 7
1160     GET@ ( X*8 + XX , 0 ) - ( X*8 + XX , 7 ) ,TEMP , G
1170     PATTERN( X , XX ) = ASC( LEFT$( MKI$( TEMP( 0 ) ) , 1 ) )
1180     PUT@ ( X*8 + XX , 8 ) - ( X*8 + XX , 15 ) , TEMP , PSET
1190   NEXT XX
1200 NEXT X
1210 ´
1220 ´begin to display
1230 PRINT "ready for display!"
1240 PRINT "start ?"
1250 K$ = "" : WHILE K$ <> "y" AND K$ <> "Y" : K$ = INKEY$ : WEND
1260 ´display loop
1270 WHILE 1
1280 BEEP 1 : FOR I = 0 TO 10 : NEXT I : BEEP 0 ´mark of start
1290 FOR I = 0 TO 50 : NEXT I
1300 FOR X = 0 TO LEN(A$) - 1
1310   FOR XX = 0 TO 7
1320     POKE &HFDFE , PATTERN( X , XX )
1330     FOR I = 0 TO 2 : NEXT I
1340   NEXT XX
1350 NEXT X
1360 POKE &HFDFE , &H00
1370 FOR I  = 0 TO 100 : NEXT I
1580 BEEP 1 : FOR I = 0 TO 10 : NEXT I : BEEP 0 ´mark of start
1590 FOR I = 0 TO 50 : NEXT I
1600 FOR X = LEN(A$) - 1TO 0 STEP -1
1610   FOR XX = 7 TO 0 STEP -1
1620     POKE &HFDFE , PATTERN( X , XX )
1630     FOR I = 0 TO 2 : NEXT I
1640   NEXT XX
1650 NEXT X
1660 POKE &HFDFE , &H00
1670 FOR I  = 0 TO 100 : NEXT I
1680 WEND
```

□のFOR～NEXTループの回数を変えて，タイミングを調整してください。

FMシリーズマニュアルバグ情報

「FM-7ユーザーズマニュアルシステム解説」編

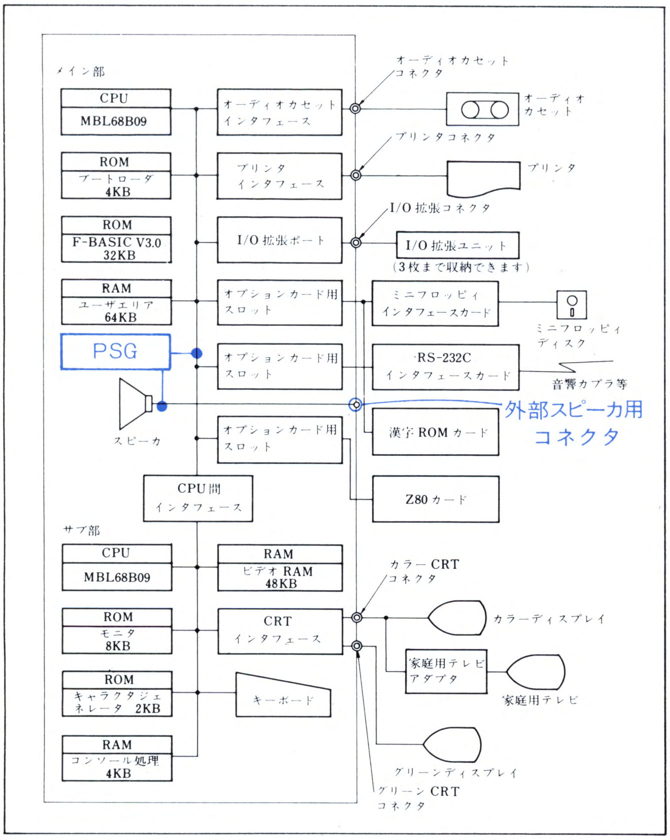
FMシリーズのマニュアル類は，かなり詳しく書かれており，ユーザーにはありがたいことです。しかし，そのマニュアルにバグがあると，初心者はまごついてしまいます。ある程度わかっている人にはなんでもない誤植でも，はじめてパソコンにさわった人は，なにがなんだかわからなくなってしまうこともあります。

そこで，今回は「FM-7ユーザーズマニュアルシステム解説」のバグ情報をお知らせします。ただし，だれが見ても明らかに誤植とわかるものは省きました。

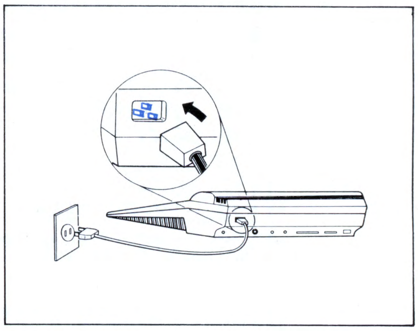
これからFM-7を買おうという人も，マニュアルを手にしたら，まずこのバグ情報をチェックしてみてください。

なお，青色部分が訂正部分です。

P.1-3　1.1.2　本体ブロック図

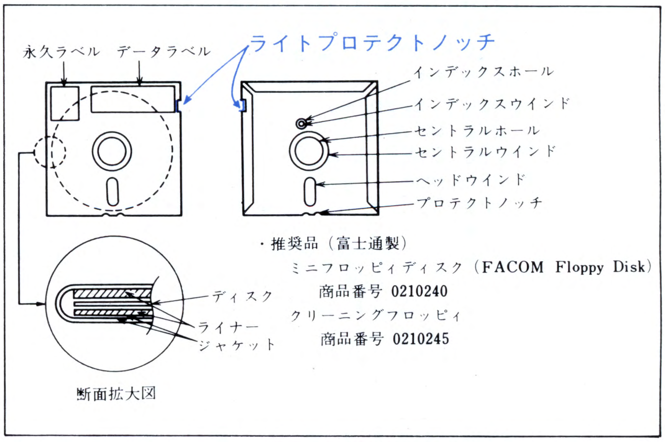


P.3-4



(注：P.3−5の電源用コネクタの図も同様です。)

P.3-26



・推奨品（富士通製）
ミニフロッピィディスク（FACOM Floppy Disk）
商品番号 0210240
クリーニングフロッピィ
商品番号 0210245

P.2-2

②カラーCRTディスプレイの解像度 ~~320~~ 640×200ドット

④グリーンCRTディスプレイの解像度 ~~320~~ 640×200ドット

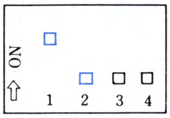
P.3-2

b ……~~ヒューズ~~ グラフィックパターン用シール

(注：下の絵もグラフィックパターン用シールになります。)

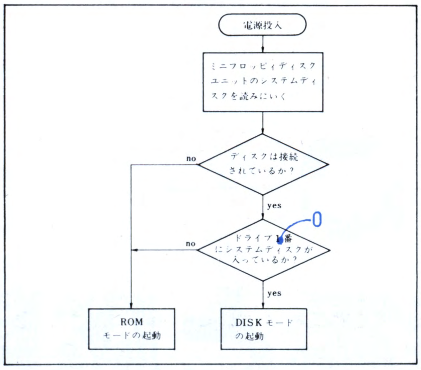
P.4-2

(2) DOSモード

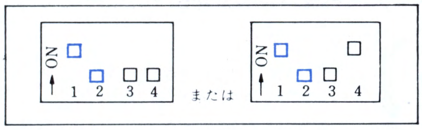


1番を~~OFF~~ ON，2番を~~ON~~ OFF，3番をOFFに設定し，

P.4-3



P.4-7



P.5-13 表組

~~RF~~ PFキー	DISKモード
PF1	AUTO␣ → なし
~~RF8~~ PF8	FILES なし ⏎

P.6-1 AUTO(オート)コマンドによる方法

PF1 A U T O ⏎ （ディスクモードの場合）

F-BASIC のコマンドに AUTO コマンドがありますが，これは，プログラムを入力する際に自動的に行番号を発生させる働きをします．

一般的にコマンドを入力するには，各アルファベット（この場合は，AUTO）をキー入力しますが，FM-7 では，このコマンドを PF1 キーに，定義してありますので，PF1 キーを押すことで，

A U T O

と，キー入力したと同じ状態になります．

PF1 のつぎの“␣”（スペース）は，1文字あけることを意味し，キーボード手前の8キー分の長さを持つキーを1回押しますが，（ROMモードでは）“AUTO␣”まで，PF1 キーに定義されておりますので，この場合は，キー入力はしません．

1 0 のキー入力は，「行番号の始まりを10にする」という意味です．1 , 0 は，キーボード（右側の）テンキーのもの，フルキーの中の 1 ヌ，0 ワ を キー入力，いずれの場合も同じで入力結果が得られます．

⏎（リターンキー）がキー入力されると，CPU は，AUTO コマンドの実行を行ないます．リターンキーも数字キーと同様に，テンキー，フルキーのいずれのキー入力も同じ働きをします．

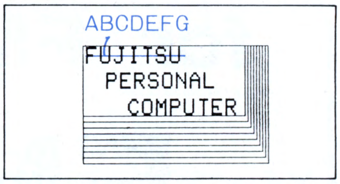
P.6-2 下3行

（ ）の入力方法：SHIFT キーを押しながら 8 (，9) を押す．

，の入力方法：フルキー部の ，ネ または，テンキー部の ， を押す．

－の入力方法：フルキー部右上部の －ホ ~~を~~ 押す。，またはテンキー部の － を

P.6-8 (真ん中の図)



P.6-10 4行目

…行番号110「(~~175~~ 75, 45)」

P.6-11 15行目から

~~カーソルを1つ進めて，~~（削除）そこで －ホ をキー入力します．カーソルを 3 つ右に進めて，6 ．4 の順にキー入力します．

つづいて，行番号 ~~175~~ 110 の“1”を削除

P.6-15 10行目

表示すると ~~左~~ 上 図のように…

P.6-17 2行目

PF1 AUTO ~~␣~~ （削除）

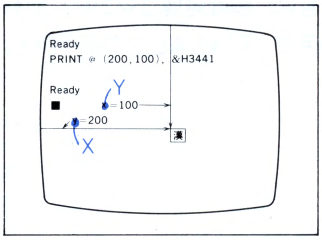
P.7-4 20行目

・40行の「PAINT(320, ~~00~~ 100), 2, 2」は…

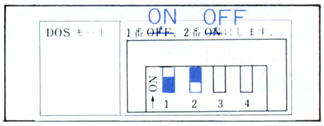
P.8-6 4行目

40 PLAY “04CGGG”, “R05CCC”, “R05EEE”

P.9-2



P.10-1 表組



P.10-3 21, 22行目

プログラム作成上，“~~LLIST~~ LPRINT”を“L？”と入力……

すると，“~~LLIST~~ LPRINT”と表示……

P.12-5 16, 17行目

~~また，このように終了……を使用して下さい。~~ 削除

P.12-15 9行目(200行の説明)

画面の ~~右~~ 左 上へファイル名を表示させます。

P. 12-31下から2行目

アスキー形式だけで~~なく，バイナリ形式でもかまいません。~~ でなければなりません。

P.12-34

〔例Ⅰ〕 すべて削除

P.13-55下から3行目

READY ？ Y ⏎

for FM-7,8,11

FMCALC――その基本機能と応用(2)

(株)ソフトマート
岡部　正

FMCALCはFMシリーズ用に作られた作表用の簡易言語である。BASICのような難しいプログラムを作らずとも，簡単に売上表，仕入表などの表組ができるソフトで，FM-7，8，11用があり，FM-11のEX，AD本体購入時にはこのソフトが付いてくる。

前号では，基本動作と，初期化コマンド，編集コマンド，フォーマットコマンドを紹介した。今号は，画面処理コマンド，計算コマンド，入出力コマンドなどのほか，FMCALCの応用例を，紹介してみよう。

3

FMCALCのコマンド(つづき)

(4)　画面処理コマンド

SORT

これは指定した行，または列に基づいて（KEY行，KEY列と呼ぶ），行，列を並べかえるコマンドである。例をとって説明しよう。

図5は，商品1～17のそれぞれについて1月～6月の売上と合計を記録したものである。SORTコマンドにより合計列（H列）をKEYとして行を数値の大きい順に並べかえると**図6**のようになる（範囲はA1，H17）。

TOTAL行を除いて行の並べかえが行われ，合計列の大きい順になっていることがわかる。本例では大きい順だが小さい順にすることもできるし，文字の場合のアルファベット，五十音順の並べかえも可能である。

実行に必要なパラメータはコマンドにあらかじめ指定するのではなく，リターンキーを押すことによってシステムが表示するメッセージに応答する形式をとっている。そのため使いやすい。

このコマンドはFMCALCの中でもひときわ有力な機能で，データ整理・分析に威力を発揮する。

数値列（行）をKEYにする場合は日付順の並べかえもできる。またSORTの結果を利用してメジアンを計算することも容易である。文字データ列（行）をKEYにした場合，名簿作成などに効果がある。ランダムに入力したものを五十音順にSORTすれば，そのまま名簿が作成できる。

LOCK

先述したように画面に現れるのは表の一部である。表示されていないエリアを参照したいとき，このコマンドが有用である。

スクロール（上下，左右）や座標指定では，ヘッダの行・列も隠れてしまう。これを避けるためにLOCKコマンドで，残したい行・列（または両方）を指定できる。

表全体が**図7**のようになる場合，画面に現れるのは**図8**の部分だ。ここでITEM14～30のデータを7月～12月の合計とともに参照したい場合，最上行と最左列をLOCKコマンドで固定しておけば**図9**のようになり，見やすくなる。

(5)　計算コマンド

#

FMCALCの計算モードには#ON（詳しくは#ONと##ONの2種類にわかれるがここでは説明は略する）と#OFFがある。前者はデータ入力と同時に入力した式の計算を行い，結果をブロックに表示させる。そのため時間がかかってしまう。

これに対して後者はデータ入力時には計算を行わないので，とくに大量のデータを入力するときは作業が迅速になる。

#コマンドにはこれらの機能のほか，式を実行させる機能（単に#と指定）がある。計算実行コマンド#は現在の計算モードが#ON，#OFFいずれでも有効である。

(6)　入出力コマンド

図5

列名	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ
		1月	2月	3月	4月	5月	6月	TOTAL
1	ITEM-1	1,849	1,184	2,189	1,119	2,239	1,189	9,769
2	ITEM-2	1,858	1,274	2,553	2,738	2,536	1,678	12,637
3	ITEM-3	1,648	1,637	2,890	2,189	1,367	1,897	11,628
4	ITEM-4	1,729	1,839	2,146	1,738	1,778	1,354	10,584
5	ITEM-5	1,984	1,803	2,782	1,980	1,678	1,245	11,472
6	ITEM-6	1,672	1,279	3,125	1,728	1,784	1,787	11,375
7	ITEM-7	2,849	1,280	3,267	1,738	1,568	1,870	12,572
8	ITEM-8	2,837	1,893	3,356	1,849	2,768	1,678	14,381
9	ITEM-9	2,610	1,782	3,792	3,783	2,749	1,426	16,142
10	ITEM-10	2,163	1,042	1,783	2,849	2,417	1,674	11,928
11	ITEM-11	2,849	1,801	1,899	2,378	2,416	1,637	12,980
12	ITEM-12	2,910	2,673	1,893	2,748	2,933	1,782	14,939
13	ITEM-13	3,910	2,892	2,536	2,539	2,633	1,935	16,445
14	ITEM-14	3,810	2,183	2,416	2,991	3,527	1,637	16,564
15	ITEM-15	2,380	2,463	2,809	3,820	3,268	2,163	16,903
16	ITEM-16	2,801	2,749	2,672	3,126	3,527	1,468	16,343
17	ITEM-17	3,289	2,138	2,781	3,729	3,268	1,673	16,878
18								
行番号 19	TOTAL	43,148	31,912	44,889	43,042	42,456	28,093	233,540

図6

	1月	2月	3月	4月	5月	6月	TOTAL
ITEM-15	2,380	2,463	2,809	3,820	3,268	2,163	16,903
ITEM-17	3,289	2,138	2,781	3,729	3,268	1,673	16,878
ITEM-14	3,810	2,183	2,416	2,991	3,527	1,637	16,564
ITEM-13	3,910	2,892	2,536	2,539	2,633	1,935	16,445
ITEM-16	2,801	2,749	2,672	3,126	3,527	1,468	16,343
ITEM-9	2,610	1,782	3,792	3,783	2,749	1,426	16,142
ITEM-12	2,910	2,673	1,893	2,748	2,933	1,782	14,939
ITEM-8	2,837	1,893	3,356	1,849	2,768	1,678	14,381
ITEM-11	2,849	1,801	1,899	2,378	2,416	1,637	12,980
ITEM-2	1,858	1,274	2,553	2,738	2,536	1,678	12,637
ITEM-7	2,849	1,280	3,267	1,738	1,568	1,870	12,572
ITEM-10	2,163	1,042	1,783	2,849	2,417	1,674	11,928
ITEM-3	1,648	1,637	2,890	2,189	1,367	1,897	11,628
ITEM-5	1,984	1,803	2,782	1,980	1,678	1,245	11,472
ITEM-6	1,672	1,279	3,125	1,728	1,784	1,787	11,375
ITEM-4	1,729	1,839	2,146	1,738	1,778	1,354	10,584
ITEM-1	1,849	1,184	2,189	1,119	2,239	1,189	9,769
TOTAL	43,148	31,912	44,889	43,042	42,456	28,093	233,540

ここではディスケットとのファイルのやりとりに関するコマンドと，印字出力に関するコマンドについて説明する。

SAVE

画面編集した表をディスケットに格納するときに用いる。SAVEしたい領域は左上と右下のブロック座標により指定する。

SAVEの単位をファイルと呼び名前をつける。必要なパラメータはシステムが表示するメッセージに応答して指定すればよい。

LOAD

SAVEとは逆にディスケットに格納されたファイルを，コンピュータに呼び出すときに用いる。呼び出したファイルをFMCALCの表に挿入する場所は自由に指定できる。たとえばA1，C10の範囲でセーブしたファイルをロードするとき，挿入場所をR8と指定しておけばR8，T17のエリアにロードされる。

オプションによりデータのみをロード，すなわち式やフォーマットなどはロードしないことも可能で，同じデータから別の計算処理をしたいときに有力な式器となる。

SAVEとLOADの組み合わせで多彩な応用を実現できるが，それについては「4．FMCALCの応用」で触れる。

PRINT

このコマンドは現在編集中の表の一部，または全部を印字出力させる。印字範囲はパラメータにより指定するが，指定エリアの幅がプリンタの行幅より大きいと，エラーメッセージが出る。

オプションにより縮小文字による印字が可能となるほか，画面表示されたエリアのみの印字も可能である。

FX

指定した範囲内のブロックに定義された式をすべて印字するコマンドである。式のレビューをする際に便利なものだ。

（7） その他のコマンド

図7

	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	TOTAL
ITEM-1	1,849	1,184	2,189	1,119	2,239	1,189	2,278	1,367	2,738	1,738	2,193	2,183	22,266
ITEM-2	1,858	1,274	2,553	2,738	2,536	1,678	2,739	1,896	1,738	1,923	2,237	3,820	26,990
ITEM-3	1,648	1,637	2,890	2,189	1,367	1,897	2,783	1,737	1,836	2,361	2,374	2,673	25,392
ITEM-4	1,729	1,839	2,146	1,738	1,778	1,354	2,738	1,839	1,725	2,263	2,294	2,740	24,183
ITEM-5	1,984	1,803	2,782	1,980	1,678	1,245	1,830	1,893	1,783	2,836	2,134	1,947	23,895
ITEM-6	1,672	1,279	3,125	1,728	1,784	1,787	1,830	1,357	1,728	2,671	2,247	1,836	23,044
ITEM-7	2,849	1,280	3,267	1,738	1,568	1,870	2,739	1,638	1,942	2,472	2,943	3,172	27,478
ITEM-8	2,837	1,893	3,356	1,849	2,768	1,678	1,479	1,830	1,528	1,263	2,043	2,462	24,986
ITEM-9	2,610	1,782	3,792	3,783	2,749	1,426	1,179	2,482	1,782	2,630	2,310	2,562	29,087
ITEM-10	2,163	1,042	1,783	2,849	2,417	1,674	3,729	2,376	2,643	2,739	2,277	1,947	27,639
ITEM-11	2,849	1,801	1,899	2,378	2,416	1,637	2,362	3,789	2,739	2,483	3,183	3,184	30,720
ITEM-12	2,910	2,673	1,893	2,748	2,933	1,782	2,638	2,749	2,739	2,943	1,194	1,856	29,058
ITEM-13	3,910	2,892	2,536	2,539	2,633	1,935	2,489	2,473	2,135	1,932	2,274	1,987	29,735
ITEM-14	3,810	2,183	2,416	2,991	3,527	1,637	1,352	2,743	2,365	3,312	2,893	2,645	31,874
ITEM-15	2,380	2,463	2,809	3,820	3,268	2,163	1,639	3,125	1,991	3,125	2,834	3,251	32,868
ITEM-16	2,801	2,749	2,672	3,126	3,527	1,468	3,263	3,364	1,247	3,127	2,637	2,553	32,534
ITEM-17	3,289	2,138	2,781	3,729	3,268	1,673	1,739	2,463	2,153	3,529	3,312	3,364	33,438
ITEM-18	3,728	3,820	3,638	2,194	3,764	1,723	2,734	2,732	2,538	3,364	3,462	3,158	36,855
ITEM-19	1,829	2,519	2,793	2,236	3,364	1,436	2,479	2,475	2,456	3,316	3,723	3,521	32,147
ITEM-20	2,891	2,781	1,820	2,730	4,127	1,720	1,278	2,845	3,326	2,459	3,315	2,889	32,181
ITEM-21	2,891	3,281	3,892	2,143	4,273	1,830	4,387	2,740	3,257	2,639	3,849	2,758	37,940
ITEM-22	4,829	3,628	2,789	3,745	3,620	2,149	2,378	2,645	2,174	2,846	3,127	2,513	36,443
ITEM-23	3,894	2,462	2,563	2,647	3,326	3,629	2,790	2,738	2,945	3,812	2,437	1,524	34,767
ITEM-24	3,792	2,167	2,956	3,281	3,389	3,317	2,378	2,748	2,748	1,199	3,749	3,163	34,887
ITEM-25	3,893	2,749	4,892	1,830	3,536	3,289	2,409	2,638	2,639	1,803	2,998	1,993	34,669
ITEM-26	3,892	3,126	2,782	2,849	2,536	3,145	1,497	3,124	2,648	3,291	3,167	1,678	33,735
ITEM-27	3,280	2,901	2,287	2,733	2,639	3,362	2,507	2,539	2,739	1,639	3,018	2,749	32,393
ITEM-28	3,628	2,891	2,437	2,810	2,730	3,256	3,208	2,546	2,416	3,329	3,820	3,126	36,197
ITEM-29	3,528	2,785	2,783	2,346	2,783	2,178	3,894	3,163	3,216	2,183	2,840	3,183	34,882
ITEM-30	3,485	2,993	2,236	3,459	3,987	3,720	3,261	3,682	2,361	3,781	2,938	3,163	39,066
TOTAL	88,708	70,015	82,757	78,045	86,530	62,847	74,006	75,736	70,275	79,008	83,822	79,600	931,349

図8

	1月	2月	3月	4月	5月	6月	7月
ITEM-1	1,849	1,184	2,189	1,119	2,239	1,189	2,278
ITEM-2	1,858	1,274	2,553	2,738	2,536	1,678	2,739
ITEM-3	1,648	1,637	2,890	2,189	1,367	1,897	2,783
ITEM-4	1,729	1,839	2,146	1,738	1,778	1,354	2,738
ITEM-5	1,984	1,803	2,782	1,980	1,678	1,245	1,830
ITEM-6	1,672	1,279	3,125	1,728	1,784	1,787	1,830
ITEM-7	2,849	1,280	3,267	1,738	1,568	1,870	2,739
ITEM-8	2,837	1,893	3,356	1,849	2,768	1,678	1,479
ITEM-9	2,610	1,782	3,792	3,783	2,749	1,426	1,179
ITEM-10	2,163	1,042	1,783	2,849	2,417	1,674	3,729
ITEM-11	2,849	1,801	1,899	2,378	2,416	1,637	2,362
ITEM-12	2,910	2,673	1,893	2,748	2,933	1,782	2,638
ITEM-13	3,910	2,892	2,536	2,539	2,633	1,935	2,489
ITEM-14	3,810	2,183	2,416	2,991	3,527	1,637	1,352
ITEM-15	2,380	2,463	2,809	3,820	3,268	2,163	1,639
ITEM-16	2,801	2,749	2,672	3,126	3,527	1,468	3,263
ITEM-17	3,289	2,138	2,781	3,729	3,268	1,673	1,739
ITEM-18	3,728	3,820	3,638	2,194	3,764	1,723	2,734
ITEM-19	1,829	2,519	2,793	2,236	3,364	1,436	2,479

TITLE

このコマンドによって，作成中の表に20文字以内のタイトルを付けることができる。タイトルは画面右下に表示され，表（一部でも可）を印字出力すると表上の中央に，拡大文字で印字される。

タイトルはファイル名とは無関係でLOAD, SAVEのときに指定する必要はない。また随時変更できるので表を分割して印字し，それぞれに別のタイトルを付けたいときは，PRINT コマンドを実行する前にTITLEコマンドで所望のタイトルを付ければよい。

FILES

このコマンドが実行されると作表画面が一時消え，現在ディスケットに格納されているファイル名がすべて表示される。このときファイル名を入力すれば，そのファイルはディスケットから削除される。

MEMORY

メモリ残量を表示するコマンドである。大量のデータや式を入力するときには，メモリの有効的な利用が重要になる。そのためにはメモリ残量を考えながら，列数やブロック幅の調整をする必要がある。MEMORYコマンドはこのように適正規模の表を作成するうえで，有力なものである。

4

FMCALCの応用

FMCALCは先述したように，作表に必要な機能をほとんど備

図 9

←固定	7月	8月	9月	10月	11月	12月	TOTAL
ITEM-14	1,352	2,743	2,365	3,312	2,893	2,645	31,874
ITEM-15	1,639	3,125	1,991	3,125	2,834	3,251	32,868
ITEM-16	3,263	3,364	1,247	3,127	2,637	2,553	32,534
ITEM-17	1,739	2,463	2,153	3,529	3,312	3,364	33,438
ITEM-18	2,734	2,732	2,538	3,364	3,462	3,158	36,855
ITEM-19	2,479	2,475	2,456	3,316	3,723	3,521	32,147
ITEM-20	1,278	2,845	3,326	2,459	3,315	2,889	32,181
ITEM-21	4,387	2,740	3,257	2,639	3,849	2,758	37,940
ITEM-22	2,378	2,645	2,174	2,846	3,127	2,513	36,443
ITEM-23	2,790	2,738	2,945	3,812	2,437	1,524	34,767
ITEM-24	2,378	2,748	2,748	1,199	3,749	3,163	34,887
ITEM-25	2,409	2,638	2,639	1,803	2,998	1,993	34,669
ITEM-26	1,497	3,124	2,648	3,291	3,167	1,678	33,735
ITEM-27	2,507	2,539	2,739	1,639	3,018	2,749	32,393
ITEM-28	3,208	2,546	2,416	3,329	3,820	3,126	36,197
ITEM-29	3,894	3,163	3,216	2,183	2,840	3,183	34,882
ITEM-30	3,261	3,682	2,361	3,781	2,938	3,163	39,066
TOTAL	74,006	75,736	70,275	79,008	83,822	79,600	931,349

→固定

えており，ごく簡単な表から複雑な計算を含む表まで幅広く応用できる。ここではいくつかの例を紹介しよう。

(1) シミュレーション

多くのビジネス用ソフトウェアは，データ入力に基づいてシステムが内部処理を行い帳表出力をする。したがってデータ入力と帳表出力の間は一種のブラックボックスになっており，時間的にも多少の隔りがある。

ところがFMCALCは，あらかじめ式を入力しておけばデータを入力するだけで，ほぼ同時に計算結果が表示される。シミュレーションを行い，結果を比較する際，能率的である。

線型計画で有名な輸送問題を例にとってみよう。

生産地A，B，C から消費地P，Q，Rに，ある物品を輸送しているとする。ある期間内のAからPへの輸送量をx_{11}，AからQをx_{12}，AからRをx_{13}……CからRへの輸送量をx_{33}とした場合，

Aから発送される量は，$x_{11}+x_{12}+x_{13}$

Bから発送される量は，$x_{21}+x_{22}+x_{23}$

Cから発送される量は，$x_{31}+x_{32}+x_{33}$

Pが受け取る量は，$x_{11}+x_{21}+x_{31}$

Qが受け取る量は，$x_{12}+x_{22}+x_{32}$

Rが受け取る量は，$x_{13}+x_{23}+x_{33}$

となる。

輸送料は輸送量に比例すると考え，輸送料を最小にするための各輸送量を求める。数学的には条件さえ設定しておけば解けるが，実際にはなかなか計算どおりにいかない。むしろシミュレーション的に解く方が現実的な場合がある。

ここで次のような表を考える（**表2**）。

X，Y，Zは各行の和，L，M，Nは各列の和であり，X＋Y＋Z＝L＋M＋Nであることが必要である。これらをそれぞれ，発送量，受入量という数値的に想定しやすい量であることから，そのうち5個を独立変数として扱う。Sは輸送料で，$\Sigma C_{ij}X_{ij}$で表せ，C_{ij}は既知とする。

9個の独立変数をx_{11}，x_{21}，x_{12}，x_{22}，X，Y，Z，L，Mとすると，ほかの変数は次のように表せる。

$x_{13}=X-x_{11}-x_{12}$

$x_{23}=Y-x_{21}-x_{22}$

$x_{31}=L-x_{11}-x_{21}$

$x_{32}=M-x_{12}-x_{22}$

$x_{33}=Z-x_{31}-x_{32}=Z-L-M+x_{11}+x_{12}+x_{21}+x_{22}$

そこでFMCALCを用いてA1，D5の範囲に上の式をあてはめると，

A1　データ
B1　データ
C1　+D1−A1−B1
D1　データ
A2　データ
B2　データ
C2　+D2−A2−B2
D2　データ
A3　+A4−A1−A2
B3　+B4−B1−B2
C3　+D3−A4−B4+A1+B1+A2+B2
D3　データ
A4　データ
B4　データ
C4　+SUM（D1，D3）−A4−B4
A5　$+x_{11}$＊A1$+x_{12}$＊B1$+x_{13}$＊C1$+x_{21}$＊A2＋x_{22}＊B2$+x_{23}$＊C2$+x_{31}$＊A3$+x_{32}$＊B3＋x_{33}＊C3

となる。X，Y，Z，L，Mは比較的予測しやすいデータなので，まずD1，D2，D3，A4，B4にデータを代入する。次に実情を考えながらA1，B1，A2，B2にいろいろな組み合わせでデータを入力し，A5の計算結果ができるだけ小さくなるように試みればよい。

表2　線型計画シミュレーション

x_{11}	x_{12}	x_{13}	X
x_{21}	x_{22}	x_{23}	Y
x_{31}	x_{32}	x_{33}	Z
L	M	N	
S			

表3

5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	4月	5月
					Ⓐ							Ⓑ

（2）　月次更新される表（SAVEとLOADの活用）

たとえば過去12か月のデータに基づいて，当月の予測値を算出する計画表は，販売・生産などあらゆる分野について必要なものである。

表3において最右欄が当月予測のエリアで，これは推定値データを入力するブロックと，Ⓐエリアのデータ，Ⓑエリアの推定データから計算されるブロックから成り立っている。FMCALCでこの表をA1，M50の範囲で作成したとしよう。

当年5月が終了した段階でこの表は更新しなければならない。単に昨年5月の欄（A列）をDELETEコマンドによって削除するだけでは予測式がL列に移ってしまい不適当である。しかし次のような手順を踏めば容易に月次更新を行うことができる。

①　あらかじめ列幅，ブロックの数値用フォーマット，式のみを定義した表に，適当なファイル名を付けてセーブしておく（たとえばFILE－0）。

②　①でセーブにしておいたファイルをA1，M50の範囲に呼び出す。Ⓐエリアにデータを埋めると同時に，Ⓑエリアに必要な推定データを入力する。これによりⒷエリアで各種予測値が計算される。この表は①の表とは別のファイル名でセーブしておく（たとえばFILE－5）。

③　当月末に②でセーブしたファイル（FILE－5）を同じ範囲（A1，M50）にロードし，DELETEコマンドによりA列を削除する。

④　③で作成したA1～L50の表をセーブする。この場合②で作成した表(FILE－5)の保存が不必要ならば，同じファイル名でもよい。

⑤　①でSAVEした表（FILE－0）をA1，M50の範囲にロードする。

⑥　④でセーブした表をA1，L50の範囲に，Dオプション付きでロードする。D指定のロードはデータのみのロードであるため，昨年6月～本年5月までのデータがⒶエリアに入力される。

⑦　M列(5月)は推定データと予測計算値から成っているので，実勢と異なる場合がある。したがってそれに合わせてM列データの修正を行う。

⑧　以上により月次更新が完成される。

（3）　名簿の作成（SORTの活用）

FMCALCは名簿の作成も容易に行える。ただし，行数が128と制限があるので完全にランダムな入力では，名簿サイズに限度がある。が，最初の一字をア行，カ行，……ワ行のように分類して異なった列に分別入力すれば，名簿サイズはかなり大きくできる。

表4

餌の種類	豚の体重増加量（グラム）						
A	610	635	580	701	640	632	633
B	595	614	550	602	633	612	601
C	527	621	564	598	601	593	584
総平均							606

表5

$$\begin{array}{cccc} x_{11} & x_{12} & \cdots\cdots & x_{1m} \\ x_{21} & x_{22} & \cdots\cdots & x_{2m} \\ \vdots & & & \\ x_{m1} & x_{m2} & \cdots\cdots & x_{mm} \end{array}$$

たとえば名前と住所だけの五十音順の名簿作成の場合，A列は先頭文字がア行の名前，B列はA列に対応する住所，C列は先頭文字がカ行の名前，D列はC列に対応する住所……S列は先頭文字がワ行の名前，T列はS列に対応する住所，というように先頭文字だけの分類で名前，住所を入力する。

次にSORTコマンドを実行する。まず，先頭文字ア行についてはSORTの範囲はA1，B128，KEY列はA列，大きい順／小さい順の区別は小さい順である。

コマンド実行の結果，A列は第二文字以下が五十音順に完全に並べかえられ，B列の住所データもこれに従う。そこでA列，B列の印字出力をとれば，名簿の一部が完成された形で求められる。カ行，サ行……ワ行についても同様な手順でSORTとPRINTを行えば，名簿が完成する。

この方法はあるドキュメントから索引を作成する作業にも適用できる。

（4）　統計的分析

複数の要因がおよぼす影響について，各要因間に有意の差があるか否かを分析する統計的手法に，分散分析がある。分散分析では実験（調査）データを表にまとめるので，FMCALCを応用すると便利である。分散分析についての詳細な説明は省くが，具体例を一つあげよう（引用は“Statistics in Action”, Penguin Books Ltd. より）。

表4は21頭の豚を3つのグループにわけ，それぞれに異なった種類の餌を与えたときに得た，おのおのの豚の体重増加量を示したものだ。分析目的は3種類の餌について体重増加という効果に有意の差はない，という統計的仮説を検定することである。

一般的に分散分析の実験データは，**表5** ，あるいはマトリクスの形で書ける。

分散分析は次のような手順で行う。

① 次の各平均を計算する。

・行平均＝ $\bar{x}_{i.}=\frac{1}{n}\sum_{j=1}^{n}X_{ij}$ （i＝1～m）

・列平均＝ $\bar{x}_{.j}=\frac{1}{m}\sum_{i=1}^{n}X_{ij}$ （j＝1～n）

・総平均＝ $\bar{\bar{x}}=\frac{1}{mn}\Sigma X_{ij}=\frac{1}{m}\sum_{i=1}^{m}\bar{x}_{i.}=\frac{1}{n}\sum_{j=1}^{n}\bar{x}_{.j}$

② 次の各平方和を計算する。

・行間平方和＝ $S_A=n\sum_{i=1}^{m}(\bar{x}_{i.}-\bar{\bar{x}})^2$

・列間平方和＝ $S_B=m\sum_{j=1}^{n}(\bar{x}_{.j}-\bar{\bar{x}})^2$

・誤差平方和＝ $S_E=\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-\bar{x}_{i.}-\bar{x}_{.j}+\bar{\bar{x}})^2$

③ 平方和から不偏分散を求める。各式の分母は自由度である。

$V_A=\frac{S_A}{m-1}$，　$V_B=\frac{S_B}{n-1}$，　$V_E=\frac{S_E}{(m-1)(n-1)}$

④ 不偏分散比を求める。

$F_A=\frac{V_A}{V_E}$，　$F_B=\frac{V_B}{V_E}$

⑤ 仮説を検定する。

・行間に効果上の差がないとすれば，F_Aは自由度m－1,(m－1)(n－1)のF分布に従うはずなので，F_Aの値とF分布表から仮説を検定する。

・列間に効果上の差がないとすれば，F_Bは自由度n－1，(m－1)(n－1)のF分布に従うはずなので，F_Bの値とF分布表から仮説を検定する。

分散分析を頻繁に行う場合には，次のような表(**表6**)を作っておくと便利である。マトリクスの部分は一応，20行20列としておこう。

D3～W22は20行20列の実験データ入力エリア，A1は行数(m)，A2は列数（n）の入力ブロックとする。

X列は行平均で，Xi＝＋AVR（Di，Wi）　(i＝3～22)

24行は列平均で，j24＝＋AVR（j3，j22）　(j＝D～X)

X24には総平均が計算される。

Y列は平方で，Yi＝＋(Xi－X24) ^2　(i＝3～22)

26列も平方で，j26＝＋(j24－X24) ^2　(j＝D～Z)

誤差平方和の計算のため，20行21列のエリアをD28，X47の範囲にとり，D3→D28，W3→W28，D4→D29，W4→W29，D22→D47，W22→W47に対応させる。

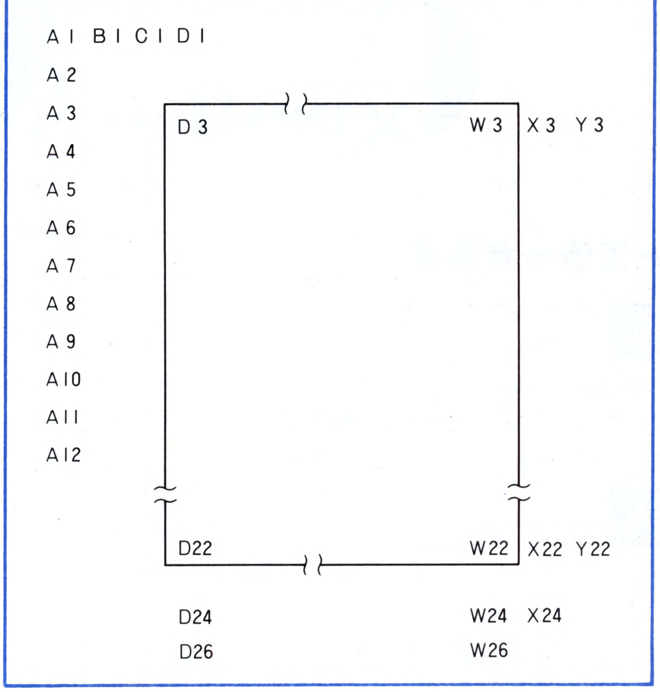
D28は＋(D3－X3(行平均)－D24(列平均)＋X24(総平均)) ^2のように平方を計算させ，X行にその和すなわちXi ＝＋SUM(Di，Wi)を計算させる（ただしi＝28～47)。

A4に不偏分散V_Aを計算させる。

A4＝＋A2＊SUM(Y3，Y22)／(A1－1)

A5に不偏分散V_Bを計算させる。

表6



A5＝＋A1＊SUM(D26，X26)／(A2－1)

A6に不偏分散V_Eを計算させる。

A6＝＋SUM(X28，X47)／(A1－1)／(A2－1)

A7に不偏分散比F_Aを計算させる。

A7＝＋A4／A6

A8，A9に自由度を計算させる。

A8＝＋A1－1，　A9＝＋(A1－1)(A2－1)

A10に不偏分散比F_Bを計算させる。

A10＝＋A5／A6

A11，A12に自由度を計算させる。

A11＝＋A2－1，　A12＝＋A9

以上の結果，A7～A12の数値とF分布表があれば，仮説の検定が可能となる。

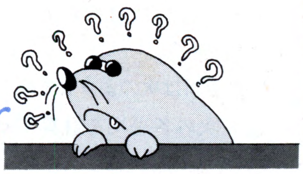
5 まとめ

以上，述べたように，FMCALCには多彩な機能が盛り込まれている。単に便利なだけでなく，ユーザーが自分で表をデザインできる点が大きな特徴である。

ビジネス用として売上管理，仕入管理，在庫管理，資金繰り，損益計算，原価計算などに利用できる。そのほか科学技術に必要なデータ処理にも応用できる機能を豊富に備えている。学校などで生徒の身体データや成績管理をすることもできよう。

そのような，いわば本格的な応用だけでなく，なにしろ電源ONだけで表の枠組が表示されデータや式の入力がREADYの状態にあるのだから，社内野球の成績管理とか，ゴルフのスコア表，家計簿の整理など，何にでも好きなものに使えるのがFMCALCの顕著な長所といえよう。

Question & Answer



▶エラーが出る

Q Oh！FM第3号34ページ ミンキーモモのプログラムで，

Illegal Function Call In 420

が出ます。全くわかりません。教えてください。(FM-8使用)（大宮市・小堀純一）

A なんと，この小堀さん，リストまで送ってくださいました。確かに420行は合っています。しかし，です。LINE文などのパラメータに，READ文で読み出した変数が使われています。この読み出したデータが違うので，このようなエラーが出るのです。そこでリストを見ると，ありました。680行の72.5です。ピリオドとカンマを打ち誤って72，5になっています。この680行については，ほかにもミスった方が多数いらしたようなので，あえて取りあげました。

なお，RAINY DAY，さがしものゲーム（第2号）でも，同じようにパラメータにREAD文で読み出した変数が使われている個所で，このエラーを出したが……という質問が多数ありました。同じようにDATA文をもう一度見直してください。

Q ワードサーチ&リプレイスプログラムをFM-11で走らせたところ，ワードサーチだけならうまく作動しますが，リプレイスワードをインプットリターンしたところ，

UNPRINTABLE ERROR

になりました。今はワードサーチだけを大変便利に使っています。ぜひプログラムの訂正個所を教えてください。（奈良県・吉岡達雄）

A FMシリーズでは，リセット後，オープンできるファイル数の最大値を聞いてきます。ここでReturnキーを押すと，最大で1ファイルしか一度にオープンできません。このプログラム，サーチでは1ファイルしか使っていませんが，リプレイスでは2ファイル，オープンします。そこで，リセット（電源投入）後，ファイル数に2以上の数をセットすれば，完全に動きます。

なお，このプログラムの作者によって，ラベルプログラムが完成しています。これは，このプログラムの原理を応用したもので，ラベルを行番号に直すものです。近々登場する予定ですが，作者によれば，「BASICでストリング処理をしたため，とても気が狂った。アセンブラに書き直そうと思っている」そうです。（編集部よりライターへ：ここにこう書いた以上，発表せざるを得まい。原稿は一刻も早く出しましょう。）

▶ノイズが出る

Q FM-8のユーザーです。8のスイッチを入れると，家中のNHKが映らなくなってしまいます（斜めのノイズが入る)。我が家にはテレビが3台あり，屋内アンテナのものは視聴不可能，屋外アンテナのものも，とても見づらくなります。この様子だと，ご近所にもかなり迷惑をかけているような気がします。

本体のどこかから，アースを引くなどして，止めることはできないでしょうか。

構成はFM-8，富士通純正グリーンディスプレイ，データレコーダ，ツクモ製倍速基板です。なお倍速基板をつける前から，ノイズは出ていました。（横須賀市・高林幹夫）

A はっきり言わせてもらうと，マイコンはノイズ発生器なんです。

アメリカなどではFCCという役所がうるさくて，この辺がしっかりしていますが，日本では取り締る規準がなく，ノイズ出し放題になっているのが現状です。

まず第一に，アースを取ってみましょう。これは本体の金属ケース（FM-8ではFG端子，FM-7では内部の金属ケース）をアースに落とせばよいのですが，近頃は，家庭で手軽にアースが取れるなんて恵まれているところは多くありません。水道管はビニール管を使っているところがほとんどですし，アースの線は「太く短く」ですから長く引きまわさなければ地面に届かないようではだめです。

それから，電源ライン。電源の線を通ってノイズが漏れることは多々あります。そこで，電源線にラインフィルタなるものを取り付けるのも効果があるでしょう。このラインフィルタ，秋葉原や日本橋の電気屋さんで売っています。近くのお店にもあるかもしれません。

そして，最終手段。部屋のまわりを厚さ10cmの鉛板でおおってしまい，その中にマイコンと発電機と酸素ボンベ，食料を持ち込みます。いざというとき，シェルターとしても使えます。いかがでしょう？

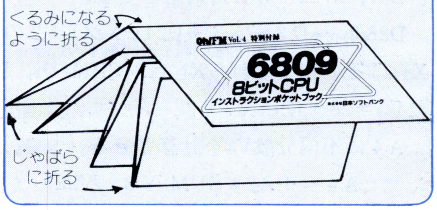
なお，隣近所へのインタフェアは，電源の漏れを除けば，まず起きません。それから，最後にもう一つ，FMはノイズ漏れの少ない機種です。

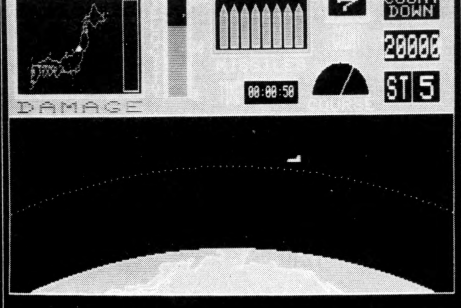
＊　＊　＊

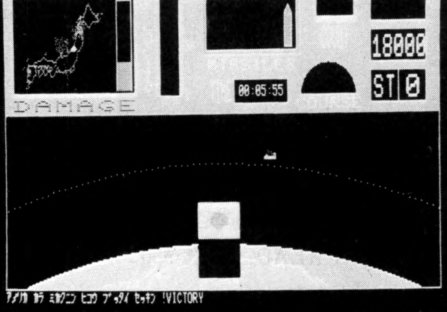
FMに関する質問をお寄せください。質問内容は文書で，できるだけ具体的にお書きください。ソフト・ハードなど幅広い質問をお待ちしています。ただし，マニュアルを読めばすぐわかるような質問はご遠慮ください。住所・氏名・電話番号などを明記のうえ，Oh！FM編集部Q&Aあてお送りください。

6809ポケットブックの折り方

次ページの6809インストラクションポケットブックは，図のように折りたたんで使ってください。







for FM-7……ゲームリスト

AMPLE SOFTWARE

「スペースシャトル」

199X年，アメリカの誇るスペースシャトル５機が，「カニカンダル星雲」から来たエイリアンにスペースジャックされてしまった。

実はシャトルには，戦闘兵器が搭載されていたのだ。アメリカは日本に協力を求め，かくしてエイリアン対アメリカ・日本の戦争が始まった。

ところがアメリカのI.C.B.M.はエイリアンの新兵器によってコントロールされ，日本に向けて発射されることになった。このため日本だけがエイリアンと戦わねばならない。

あなたが指揮をとって，限られた時間内に敵をたおすのだ。

コマンド・ルール

PF1キー：ミサイル発射コマンド。このキーを押すとミサイル発射準備にはいる。発射可能（READY）の表示が出たら，←→で打上げ方向，↑↓で初速度を指定し，リターンキーを押せば10秒後にミサイルが発射される。発射不可能(IMPOSSIBLE）の場合は修理完了を待ち，再びPF1キーを押す。修理には2000ドル使われる。

PF2キー：ミサイル買取りコマンド。ミサイル１機につき500ドルかかる。

PF3キー：衛星発射コマンド。シャトルが視界内にある場合は打上げられない。

PF4キー：PF3キーによって打上げられた衛星を爆破する。その際，シャトルが衛星爆破範囲にあればシャトルも爆破できる。

＊　　＊　　＊

15分以内に敵を５機とも破壊しなければ，タイムオーバーとなる。また，アメリカからのミサイル，シャトルからのミサイルが日本に落ちるとDAMAGEが増えるが，日本のミサイル基地（日本地図の白い部分）に落ちると交戦不能となり負け。所持金が０になっても負けとなる。

プログラムの説明

初期の予定ではよくあるシミュレーションゲームがそうであるように，表示のほとんどをデジタル化するという方向でプログラムにとりかかったんですけど，どうしても人間にとって数字というのは理解しにくいものなんですね。

そこでできる限り表示するものはアナログ化，つまり視覚的に理解できるようにしました。また，シミュレーションであるためプログラムを書いている途中で新たな事象が生れて，それを書き加えるということが多かったため，全体として“みにくい”プログラムになってしまいました。それでも大きくサブルーチン化してあるので，よく見ていただければ流れがわかると思います。

対象ハードがFM-7だけということなので“7”の特徴をフルに使っています。具体的にはタイマ割り込み，インタバル割り込み，ファンクションキー割り込み，高解像度グラフィックス，サウンド機能，という具合です。

プログラムはあくまで構造化を狙ったわけですが，サブルーチンコールをファンクションキーの割り込みによって行っているので，わかりにくい部分もあると思います。メインルーチンはなぜか２つあります。１つは全体制御用で，イニシャライズ，タイトル表示などをコントロールしています。もう１つのメインループルーチンではシャトルの動きに合わせて，ミサイルなどの動きをコントロールするという方法をとっています。すなわち，このゲームではシャトルの動きが最優先であり，あとはフラグの変化で処理する，という方法をとっています。(著者談)

メインルーチン：200～

200～290：初期設定，300～520：タイトル表示，600～660：DATA READ，700～950：メッセージ表示，1000～1240：初期画面表示

メインループルーチン：1300～

KEY(1)割り込み（ミサイル打上げ）：2000～2240

KEY(2)割り込み（ミサイル買取り）：2250～2410

KEY(3)割り込み（衛星打上げ）：2500～2580

KEY(4)割り込み（衛星爆破）：2800～2910

＊　＊　＊

この商品はアンプルソフトウェア(株)から発売されています。商品に関するお問い合わせは，アンプルソフトウェアへお願いします。ただし，プログラム内容に関するお問い合わせは，Oh! FM編集部へお願いします。

○このプログラムは，個人で利用するほかは著作権法上 無断複製を禁じられています。
COPY RIGHT © 1983 AMPLE SOFTWARE

```
0 REM ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
1 REM ■                                       ■
2 REM ■    SPACE SHUTTLE  THE COMPUTER GAME   ■
3 REM ■              Programed by Y.ITAKURA   ■
4 REM ■        Produced by AMPLE SOFTWARE     ■
5 REM ■                           MARCH 1983  ■
6 REM ■                                       ■
7 REM ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
8 REM *****************************************
9 '
10 REM ■■■■■■■■■■■■■■  MAIN ROUTINE  ■■■■■■■■■■■■■■
20 GOSUB 200:'INITIALIZE
30 IF MORE=0 THEN GOSUB 300:'TITLE
40 GOSUB 600:'DATA READ
50 IF MORE=0 THEN GOSUB 700:'Massage
60 GOSUB 1000:'ｼｮｷ ｶﾞﾒﾝ
70 GOTO 1300:'ｶﾞﾒﾝ WORK
80 END
200 REM ■■■■■■■■■■■■■  INITIALIZE  ■■■■■■■■■■■■■
210 WIDTH 80,25:CONSOLE 0,25,0,0:COLOR 7,0:CLS:TIME$="00:00:00"
220 DIM Z%(550),S%(20),M%(60),E%(10)
230 RANDOMIZE(VAL(RIGHT$(TIME$,2)))
240 ON KEY(1) GOSUB 2000:ON KEY(2) GOSUB 2250:ON KEY(3) GOSUB 2500:ON KEY(4) GOS
UB 2800:ON TIME GOSUB 4310:TIME "00:15:00":TIME ON
250 MISSILE=8:SSTEP=.001:COL=7:IYEN=20000:PI=3.14159:SHUTTLE=5:DAMAGE=0:KOTOWARI
=0
260 FOR I=1 TO 7
270 COLOR=(I,I)
280 NEXT I
290 RETURN
300 REM ■■■■■■■■■■■■■  TITLE GRAPHIC  ■■■■■■■■■■■■■
310 FOR I=0 TO 300:X=RND(1)*639:Y=RND(1)*199:C=INT(RND(1)*6+1):PSET(X,Y,C):NEXT
I
320 FOR S=3.1415 TO 3.85 STEP .002
330 PSET(COS(S)*600+1100,-(SIN(S)*300),7)
340 NEXT S
350 CONNECT(540,0)-(540,3)-(530,10)-(535,20)-(525,25)-(520,30)-(515,34)-(510,38)
-(509,45)-(509,50)-(518,52)-(518,55)-(530,55)-(540,60)-(550,65)-(554,70)-(546,80
)-(550,85)-(560,90)-(565,95)-(569,105)-(575,108)-(578,128)-(585,138)-(600,145)-(
595,140),7
360 CONNECT(595,140)-(590,130)-(600,132)-(608,140)-(610,145)-(625,155)-(639,160)
,7:PAINT(520,1),1,7:PAINT(530,60),1,7:PAINT(580,1),4,7
370 FOR S=3.1415 TO 3.8 STEP .005
380 PSET(COS(S)*620+RND(1)*25+1100,-(SIN(S)*310),5)
390 NEXT S:GOSUB 4100:GOSUB 3930
400 CONNECT(330,128)-(125,106)-(110,107)-(96,114)-(60,120)-(44,124)-(37,127)-(36
,130)-(37,134)-(46,139)-(80,147)-(310,176)-(380,182)-(496,182)-(498,181)-(496,18
0)-(305,161)-(294,155)-(280,154)-(140,145)-(140,130)-(100,130)-(98,123)-(88,119)
-(78,117),7
410 CONNECT(100,126)-(80,126)-(75,123)-(72,122)-(60,120),7:CONNECT(330,128)-(438
,79)-(466,79)-(468,81)-(467,83)-(420,125)-(410,127)-(406,129)-(420,131)-(438,142
)-(445,142)-(447,150)-(440,151)-(438,152)-(390,154)-(380,152)-(365,148)-(168,127
)-(140,130),7
```

```
420 CONNECT(330,128)-(408,128),7:CONNECT(108,109)-(125,110)-(140,115)-(140,118)-
(130,118)-(128,118)-(115,115)-(96,113),7:CONNECT(390,154)-(415,156)-(415,171),7:
CONNECT(365,148)-(374,146)-(376,140)-(374,134)-(360,131)-(346,128),7
430 CONNECT(346,128)-(359,133)-(364,140)-(364,148),7:CONNECT(376,140)-(400,145)-
(420,146)-(439,145),7:CONNECT(439,142)-(439,150),7
440 CONNECT(348,130)-(349,124)-(436,83)-(450,83)-(400,118)-(400,123)-(392,128),7
:CONNECT(462,88)-(452,88)-(409,120)-(422,123),7:PAINT(258,128),7,7:PAINT(410,100
),7,7:PAINT(439,104),7,7
450 CONNECT(218,132)-(218,122)-(215,119)-(206,115),0:CONNECT(166,127)-(166,117)-
(162,113)-(150,109),0:CONNECT(266,136)-(266,128)-(265,125)-(254,120),0:CONNECT(3
14,142)-(314,134)-(315,130)-(305,126),0:PAINT(400,150),7,7:CONNECT(496,180)-(270
,163)-(190,153)
460 FOR R=35 TO 10 STEP -5:CIRCLE(412,163),R,7,.3,.8,.18:NEXT R
470 CONNECT(140,145)-(138,149)-(128,153),7
480 FOR I=1 TO 240
490 SOUND 2,I+1:SOUND 4,I+2:SOUND 0,I:SOUND 6,I:SOUND 8,15:SOUND9,15:SOUND10,15:
SOUND 7,4:NEXT I
500 SYMBOL(30,10),"SPACE",4,5,4:FOR I=0 TO 3000:NEXT I:SYMBOL(240,10),"SHUTTLE",
4,5,4
510 FOR I=0 TO 2000:NEXT I:SYMBOL(190,53),"THE COMPUTER GAME",2,2,1
520 RETURN
600 REM ██████████ DATA READ ██████████
610 GOSUB 3540'JAPAN Data read
620 GOSUB 3750'SPST Data read
630 GOSUB 3780'MISSILE data read
640 SYMBOL(50,180),"HIT ANY KEY",2,2,3
650 IF INKEY$="" THEN 650
660 RETURN
700 REM ██████████ GAME MASSAGE ██████████
710 CLS:WIDTH80,25:CONSOLE0,25,0,0
720 SYMBOL(150,25),"SPACE SHUTTLE",3,5,1:GOSUB 4270
730 M2$=" Space shuttle ,which was constracted for warning, is flying and flying
 !!"
740 M3$=" I.C.B.M.(Intercontinental Ballistic Missile) is approaching and coming
 !!"
750 M4$=" We have only 8 missiles ,$20000 ,a few weapon and little time."
760 M5$=" You must destroy 5 shuttles for 15 minutes."
770 LOCATE2+W,9:FOR W=1 TO LEN(M2$)
780 A$=MID$(M2$,W,1):IF A$<>" " THEN PLAY"L64O6C"
790 LOCATE1+W,9:PRINT"■";:LOCATE1+W,9:PRINT A$;:FOR T=0 TO 10:NEXT T
800 NEXT W:FOR T=0 TO 500:NEXT T
810 LOCATE2+W,11:FOR W=1 TO LEN(M3$)
820 A$=MID$(M3$,W,1):IF A$<>" " THEN PLAY"L64O6C"
830 LOCATE1+W,11:PRINT "■":LOCATE1+W,11:PRINT A$;:FOR T=0 TO 10:NEXT T
840 NEXT W:FOR T=0 TO 500:NEXT T
850 LOCATE2+W,13:FOR W=1 TO LEN(M4$)
860 A$=MID$(M4$,W,1):IF A$<>" " THEN PLAY"L64O6C"
870 LOCATE1+W,13:PRINT"■":LOCATE1+W,13:PRINT A$;:FOR T=0 TO 10:NEXT T
880 NEXT W:FOR T=0 TO 500:NEXT T
890 LOCATE2+W,15:FOR W=1 TO LEN(M5$)
900 A$=MID$(M5$,W,1):IF A$<>" " THEN PLAY"L64O6C"
910 LOCATE1+W,15:PRINT"■":LOCATE1+W,15:PRINT A$;:FOR T=0 TO 10:NEXT T
920 NEXT W:FOR T=0 TO 500:NEXT T
930 COLOR 4,0:LOCATE24,20:PRINT "HIT ANY KEY FOR GAME START"
940 IF INKEY$="" THEN 940
950 RETURN
1000 REM ██████████ ｼｮｷ ｶﾞﾒﾝ ｻｸｾｲ ██████████
1010 GOSUB 4270:WIDTH 80,25:CONSOLE 24,1,0,0:COLOR 7,0:CLS
1020 LINE(0,0)-(639,190),PSET,7,B:LINE(0,81)-(639,81),PSET,7:LINE(10,5)-(200,70)
,PSET,7,B:LINE(13,7)-(197,68),PSET,7,B:PAINT(12,6),6,7:LINE(230,5)-(270,70),PSET
,7,B:LINE(235,8)-(265,70),PSET,7,B:PAINT(233,15),3,7:LINE(300,5)-(440,45),PSET,7
,B
1030 LINE(230,5)-(235,8),PSET,7:LINE(265,8)-(270,5),PSET,7:LINE(305,8)-(435,42),
PSET,7,B:PAINT(302,10),4,7:LINE(170,10)-(192,66),PSET,7,B
1040 LINE(540,5)-(620,25),PSET,7,B:LINE(540,30)-(620,50),PSET,7,B:LINE(540,55)-(
620,75),PSET,7,B:LINE(580,55)-(580,75),PSET,7:LINE(460,5)-(520,25),PSET,7,B:LINE
(345,60)-(422,75),PSET,7,B
1050 LINE(440,70)-(520,70),PSET,7:CIRCLE(480,70),41,7,.52,.5,0:PAINT(320,1),1,7:
```

```
SYMBOL(433,72),"COURSE",2,1,7
1060 LINE(463,8)-(517,22),PSET,7,B:PAINT(462,10),5,7
1070 PUT@(20,7)-(156,67),Z%,PSET,4:CIRCLE(105,40),4,7,,,,F
1080 SYMBOL(10,72),"DAMAGE",4,1,2:X=210:SYMBOL(X,7),"V",2,1,7:SYMBOL(X,15),"E",2
,1,7:SYMBOL(X,23),"L",2,1,7:SYMBOL(X,31),"O",2,1,7:SYMBOL(X-1,39),"C",2,1,7:SYMB
OL(X-1,47),"I",2,1,7:SYMBOL(X-1,55),"T",2,1,7:SYMBOL(X-1,63),"Y",2,1,7
1090 LINE(268,40)-(272,40),PSET,7:LINE(268,10)-(272,10),PSET,7:SYMBOL(271,6),"10
0",1,1,7:SYMBOL(273,36),"50",1,1,7:SYMBOL(273,66),"0",1,1,7:SYMBOL(306,48),"MISS
ILES",2,1,7:SYMBOL(470,28),"COUNT",1,2,7:SYMBOL(310,61),"TIME",1,2,7
1100 SYMBOL(525,34),STR$(IYEN),2,2,7:SYMBOL(544,59),"ST",2,2,7:SYMBOL(565,59),ST
R$(SHUTTLE),3,2,7
1110 FOR IM=313 TO 418 STEP 15:PUT@A(IM,14)-(IM+10,40),M%,PSET:NEXT IM
1120 'EARTH
1130 FOR S=1.2 TO 1.94 STEP .001
1140 PSET(COS(S)*800+320,-(SIN(S)*400)+562,7):NEXT S
1150 CONNECT(350,162)-(348,165)-(347,167)-(341,167)-(341,168)-(345,169)-(340,172
)-(332,174)-(335,174)-(345,172)-(360,171)-(364,172)-(375,172)-(380,170)-(385,170
)-(390,174)-(410,175)-(438,176)-(440,179)-(450,183)-(460,188)-(462,188)-(461,186
),7
1160 CONNECT(458,185)-(458,183)-(465,185)-(475,188)-(483,190),7:CONNECT(305,162)
-(305,164)-(312,164)-(312,165)-(316,166)-(308,168)-(300,167)-(290,167)-(284,170)
-(276,174)-(272,175)-(266,175)-(262,174)-(264,171)-(272,168)-(268,168)-(260,170)
-(240,171),7
1170 CONNECT(240,171)-(225,171)-(220,173)-(220,174)-(225,175)-(220,178)-(200,184
)-(192,184)-(182,185)-(170,185)-(170,188)-(165,189)-(160,187)-(160,187)-(150,187
)-(140,190),7
1180 CONNECT(246,173)-(242,174)-(235,177)-(232,178)-(242,177)-(246,175)-(246,173
),7
1190 CONNECT(232,179)-(228,181)-(224,180)-(221,182)-(224,182)-(229,183)-(232,182
)-(237,182)-(232,181)-(232,179),7
1200 CONNECT(222,182)-(210,186)-(190,186)-(190,187)-(200,188)-(202,189)-(215,188
)-(220,181)-(222,182),7
1210 CONNECT(190,190)-(188,188)-(180,188)-(180,190),7:PAINT(320,170),1,7:PAINT(1
40,180),4,7:PAINT(480,180),4,7:PAINT(210,186),4,7:PAINT(182,189),4,7:PAINT(204,1
87),4,7:PAINT(229,182),4,7:PAINT(154,189),1,7:PAINT(241,176),4,7:PAINT(224,181),
4,7
1220 FOR S=1.2 TO 1.94 STEP .01
1230 PSET(COS(S)*900+320,-(SIN(S)*450)+562,6):NEXT S
1240 TIME$="00:00:00":SOUND 8,0:SOUND 9,0:SOUND 10,0:SOUND 7,56:RETURN
1300 REM [redacted] MAIN LOOP ROUTINE [redacted]
1310 KEY(1) ON:KEY(2) ON:KEY(3) ON:KEY(4) ON
1320 X=COS(1.3)*920+310:Y=-(SIN(1.3)*460)+562
1330 FOR S=1.25 TO 1.91 STEP SSTEP:'Main roop
1340   IF F=1 THEN GOSUB 2920:'Missile move
1350    IF UFO=1 THEN GOSUB 1750:'UFO move
1360     LINE(X,Y)-(X+20,Y+7),PSET,0,BF
1370      X=COS(S)*920+310:Y=-(SIN(S)*460)+562:PUT@(X,Y)-(X+18,Y+6),S%,PSET,COL
1380      IF INT(X)=210 AND COL=7 THEN GOSUB 1600
1390      IF SPO=1 THEN GOSUB 1700
1400     IF F=0 AND KFLUG<>1 THENKEY(1) ON:KEY(3) ON ELSE KEY(1) OFF:KEY(3) OFF
1410    IF UFO=0 THEN GOSUB 1500:'ICBM
1420   GOSUB 3470:'Time counter
1430 NEXT S
1440 LINE(X,Y)-(X+18,Y+6),PSET,0,BF
1450 IF COL=0 THEN COL=7 ELSE COL=0
1460 SSTEP=SSTEP+.00005
1470 GOTO 1330
1480 END
1500 REM [redacted] UFO [redacted]
1510 IF INT(RND(1)*200)=100 THEN 1520 ELSE RETURN
1520 GOSUB 4210:'ｷｹﾝ ｼﾝｺﾞｳ
1530 LOCATE0,24:PRINT "ｱﾒﾘｶ ｶﾗ ﾐｶｸﾆﾝ ﾋｺｳ ﾌﾞｯﾀｲ ｾｯｷﾝ !!!         ";
1540 UFO=1:UF=8
1550 RETURN
1600 REM [redacted] SP [redacted]
1610 GOSUB 4210
1620 LOCATE0,24:PRINT "ｼｬﾄﾙ ｶﾗ ﾐｻｲﾙ ﾗｯｶ !!!                              ";
1630 SPO=1:SPD=0:SPX=X+10
```

```
1640 RETURN
1700 REM [redacted] SP-PSET [redacted]
1710 PSET(SPX,SPY,0)
1720 SPD=SPD+1:SPY=Y+10+SPD:IF SPY>=180 THEN GOSUB 1810:SPO=0:RETURN
1730 PSET(SPX,SPY,3)
1740 RETURN
1750 REM [redacted] UFO PSET [redacted]
1760 UF=UF-.1:UFX=UF*20+380:UFY=UF^2+120
1770 PSET(UFX,UFY,6)
1780 IF UF=<-8THEN UFO=0:GOTO 1810
1790 PSET(UF*20+380,UF^2+120,0)
1800 RETURN
1810 REM [redacted] DAMAGE [redacted]
1820 DX=RND(1)*120+20:DY=RND(1)*50+7:GOSUB 4180
1830 FOR DA=0 TO 100
1840 PSET(DX+RND(1)*20,DY+RND(1)*10,2)
1850 NEXT DA
1860 DAMAGE=DAMAGE+RND(1)*50:IF DAMAGE>=200 THEN LOCATE0,24:PRINT"ニッホン チンボツ
                                        ";:PAINT(14,8),5,7:GOTO 3210
1870 IF DX>85 AND DX<105 AND DY>30 AND DY<40 THEN LOCATE0,24:PRINT "MISSILE キチ カ
" ハカイ サレマシタ !! モウ マケデス コウフク シマショウ。           ";:GOTO 3210
1880 LINE(171,66-DAMAGE/4)-(191,66),PSET,2,BF
1890 LOCATE0,24:PRINT "ニッポン ニ MISSILE ガ オチマシタ !! DAMAGE !!
      ";
1900 RETURN
2000 REM [redacted] KEY(1) MISSILE ウチアゲ [redacted]
2010 PUT@(X,Y)-(X+18,Y+6),S%,PSET,COL:KEY(1) OFF:KEY(3) OFF:KFLUG=1:LINE(541,6)-
(619,24),PSET,0,BF
2020 IF MISSILE<=0 THEN LOCATE0,24:PRINT"モウ MISSILE ハ アリマセン。
               ";:KFLUG=0:KEY(3) ON:RETURN
2030 IF UFO=1 THEN PSET(UFX,UFY,6)
2040 BEEP:LOCATE0,24:PRINT "ウチアゲ シュンビ ニ ハイリマス                          ";:F
OR W=0 TO 100:GOSUB 3470:NEXT W
2050 UCHIAGE=INT(RND(1)*10):IF UCHIAGE=5 THEN 2060 ELSE 2100
2060 SYMBOL(540,8),"IMPOSSIBLE",1,2,2:GOSUB 4210:LOCATE0,24:PRINT"ジコ ガオコリマシタ。
シュウリニ ¥2000 カカリマシタ。                          ";:KFLUG=0:KEY(3) ON
2070 IYEN=IYEN-2000
2080 LINE(541,31)-(619,49),PSET,0,BF:PSET(UFX,UFY,0):SYMBOL(525,34),STR$(IYEN),2
,2,7
2090 IF IYEN<=0 THEN 3210 ELSE RETURN
2100 BEEP:SYMBOL(540,8),"READY",2,2,4:LOCATE0,24:PRINT "ウチアゲ シュンビ カンリョウ シマシタ
                      ";:FOR I=0 TO 100:GOSUB 3470:NEXT I
2110 LOCATE0,24:PRINT"ウチアゲ ホウコウ ト ショソクド ヲ セッテイ シテ クダサイ     ";
2120 OX=480:OY=51:VY=70:SC=3.1415/2
2130 LINE(480,69)-(OX,OY),PSET,6:A$=INKEY$:IF A$=CHR$(13) THEN 2190 ELSE IF A$="
" THEN 2180
2140 IF A$=CHR$(&H1D) AND SC<3 THEN LINE(480,69)-(OX,OY),PSET,0:SC=SC+.05:OX=COS
(SC)*38+480:OY=-SIN(SC)*19+70
2150 IF A$=CHR$(&H1C) AND SC>.1 THEN LINE(480,69)-(OX,OY),PSET,0:SC=SC-.05:OX=CO
S(SC)*38+480:OY=-SIN(SC)*19+70
2160 IF A$=CHR$(&H1E) AND VY<>9 THEN VY=VY-1:LINE(236,VY)-(264,VY),PSET,2:GOTO 2
130
2170 IF A$=CHR$(&H1F) AND VY<70 THEN LINE(236,VY)-(264,VY),PSET,0:VY=VY+1
2180 GOSUB 3470:GOTO 2130
2190 'ウチアゲ ショキ セッテイ
2200 VELOCITY=(70-VY)*30
2210 SS=10:MX=220:MY=180
2220 LINE(541,6)-(619,24),PSET,0,BF:SYMBOL(540,8),"COUNT",2,1,6:SYMBOL(547,16),"
DOWN",2,1,6:LINE(X,Y)-(X+18,Y+6),PSET,0,BF
2230 IF UFO=1 THEN PSET(UFX,UFY,0)
2240 LOCATE0,24:PRINTSPC(70):ON INTERVAL GOSUB 3500:INTERVAL 1:INTERVAL ON:RETUR
N
2250 REM[redacted] KEY(2) MISSILE SUPPLY [redacted]
2260 SUPPLY=INT(RND(1)*10):IF MISSILE=>8 THEN RETURN
2270 IF SUPPLY=5 THEN NATION$="ドイツ "
2280 IF SUPPLY=1 THEN NATION$="フランス "
2290 IF SUPPLY=2 THEN NATION$="アメリカ "
2300 IF SUPPLY=3 THEN NATION$="イギリス "
2310 IF SUPPLY=4 THEN NATION$="ソビエト "
```

```
2320 IF SUPPLY=6 THEN NATION$='ﾁｮｰｺﾞｸ'
2330 IF SUPPLY=7 THEN NATION$='ｻｳｼﾞ ｱﾗﾋﾞｱ'
2340 IF SUPPLY=8 THEN NATION$='ﾐﾅﾐ ｱﾌﾘｶ '
2350 IF SUPPLY=9 THEN NATION$='ｱﾙｾﾞﾝﾁﾝ  '
2360 IF SUPPLY=10THEN NATION$='ﾎﾟｰﾗﾝﾄﾞ '
2370 YEN$=STR$((8-MISSILE)*500)
2380 LOCATE0,24:PRINT NATION$;' ｶﾗ MISSILE ｦ $$';YEN$;'ﾃﾞ ｶｲﾄﾘ ﾏｼﾀ｡          ';
2390 FOR IM=313 TO 418 STEP 15:PUT@A(IM,14)-(IM+10,40),M%,PSET:NEXT IM
2400 IYEN=IYEN-VAL(YEN$):LINE(541,31)-(619,49),PSET,0,BF:SYMBOL(525,34),STR$(IYE
N),2,2,7:MISSILE=8:IF IYEN<=0 THEN 3210
2410 RETURN
2500 REM ████████ KEY(3) ｴｲｾｲ ｳﾁｱｹﾞ ████████
2510 IF E=0 THEN KEY(3) OFF:KEY(1) OFF:KFLUG=1ELSE RETURN
2520 PUT@(X,Y)-(X+18,Y+6),S%,PSET,COL:LINE(541,6)-(619,24),PSET,0,BF
2530 IF COL=7 THEN BEEP:LOCATE0,24:PRINT 'SHUTTLE ｶﾞ ｲﾙﾉﾃﾞ ｴｲｾｲ ﾊ ｳﾁｱｹﾞﾗﾚﾏｾﾝ
                    ';:KEY(3) ON:KEY(1) ON:KFLUG=0:LINE(X,Y)-(X+18,Y+6),PSET
,0,BF:PSET(UFX,UFY,0):RETURN
2540 BEEP:SYMBOL(540,8),'READY',2,2,4:LOCATE0,24:PRINT'ｳﾁｱｹﾞ ｼﾞｭﾝﾋﾞ ｶﾝﾘｮｳ ｼﾏｼﾀ｡
                              ';:FOR W=0 TO 100:GOSUB 3470:NEXT W:LINE(541,6)-(6
19,24),PSET,0,BF
2550 SYMBOL(540,8),'COUNT',2,1,6:SYMBOL(547,16),'DOWN',2,1,6:LINE(X,Y)-(X+18,Y+6
),PSET,0,BF:PSET(UFX,UFY,0)
2560 LOCATE0,24:PRINT SPC(70):SS=10
2570 ON INTERVAL GOSUB 2600:INTERVAL 1:INTERVAL ON
2580 RETURN
2600 REM █████ ｳﾁｱｹﾞ COUNTER ｴｲｾｲ (INTERVAL) █████
2610 IF SS=0 THEN GOSUB 3860:KEY(1) ON:KEY(3) ON:KFLUG=0:GOSUB 2640
2620 IF SS=0 THEN LINE(464,8)-(516,21),PSET,0,BF:LINE(541,6)-(619,24),PSET,0,BF:
INTERVAL OFF:E=1 ELSE SS=SS-1:LINE(464,8)-(516,21),PSET,0,BF:BEEP:SYMBOL(441,12)
,STR$(SS),4,1,7
2630 RETURN
2640 REM ██████ ｴｲｾｲ PSET ██████
2650 EX=RND(1)*600+10:EY=RND(1)*10+85
2660 SYMBOL(EX,EY),'*',1,1,7
2670 E=1:KEY(4) ON
2680 IYEN=IYEN-1000:LINE(541,31)-(619,49),PSET,0,BF:SYMBOL(525,34),STR$(IYEN),2,
2,7
2690 IF IYEN<=0 THEN 3210
2700 RETURN
2800 REM █████████ KEY(4) BOMB EISEI █████████
2810 IF E=0 THEN RETURN
2820 LOCATE0,24:PRINT 'ｴｲｾｲ ﾊﾞｸﾊ !!                                          ';
2830 GOSUB4180
2840 FOR BA=0 TO 50:PSET(EX+RND(1)*20,EY+RND(1)*8,2):NEXT BA
2850 KD=SQR((X-EX)^2+(Y-EY)^2):LINE(EX,EY)-(EX+21,EY+9),PSET,0,BF
2860 IF KD>30 THEN KEY(4) OFF:KEY(3) ON:KFLUG=0:E=0:RETURN
2870 IF COL=0 THEN KEY(4) OFF:KEY(3) ON:KFLUG=0:E=0:RETURN
2880 LINE(X,Y)-(X+18,Y+6),PSET,0,BF
2890 LOCATE0,24:PRINT 'ｼｬﾄﾙ ｦ ﾊﾞｸﾊ ｼﾏｼﾀ
     ';:GOSUB 2980:COL=0
2900 E=0:KFLUG=0:KEY(4) OFF:PSET(MX,MY,0)
2910 RETURN
2920 REM ██████ MISSILE PSET ██████
2930 PSET(MX,MY,0):GR=GR+1:MX=MX+COS(SC)*10:MY=MY-VELOCITY/500+(GR/30)^2
2940 IF NOT(MY<181 AND MY>83 AND MX>1 AND MX<638) THEN LOCATE0,24:PRINT'MISSILE
has gone                    ';:F=0:GR=0:KFLUG=0:RETURN
2950 IF COL=7 AND MX>X AND MX<X+18 AND MY>Y AND MY<Y+6 GOTO 2980
2960 IF UFY<160 AND UFO=1 AND MX>UFX-6 AND MX<UFX+6 AND MY>UFY-3 AND MY<UFY+3 GO
TO 3100
2970 PSET(MX,MY,2):RETURN
2980 REM █████ HIT SPST █████
2990 GOSUB 4180:GOSUB 3140:SHUTTLE=SHUTTLE-1:LINE(581,56)-(619,74),PSET,0,BF:SYM
BOL(565,59),STR$(SHUTTLE),3,2,7:IF SHUTTLE=0 THEN 3250:'VICTORY
3000 LOCATE0,24:PRINT 'SPACE SHUTTLE ｦ ｹﾞｷﾊ ｼﾏｼﾀ               ';:F=0:GR=0:S=1.91:
COL=0:KFLUG=0
3010 RETURN
3100 REM █████ HIT UFO █████
3110 GOSUB 4180:GOSUB 3170:UFO=0:F=0:GR=0:KFLUG=0
```

```
3120 LOCATE0,24:PRINT"ICBM ｦ ｹﾞｷﾊ ｼﾏｼﾀ                              ";
3130 RETURN
3140 REM [redacted] BOMB GRAPHIC SHUTTLE [redacted]
3150 FOR B=0 TO 20:BX=RND(1)*18+X:BY=RND(1)*6+Y:PSET(BX,BY,2):NEXT B
3160 RETURN
3170 REM [redacted] BOMB GRAPHIC UFO [redacted]
3180 FOR B=0 TO 20:BX=RND(1)*8+UFX-4:BY=RND(1)*4+UFY-2:PSET(BX,BY,2):NEXT B
3190 LINE(UFX-4,UFY-2)-(UFX+4,UFY+2),PSET,0,BF
3200 RETURN
3210 REM [redacted] ﾊｲﾎﾞｸ [redacted]
3220 GOSUB 4210:LOCATE0,24:SYMBOL(20,100),"ﾜﾀｼﾀﾁﾊ  ﾊｲﾎﾞｸ ｼﾏｼﾀ",4,4,2
3230 FOR W=0 TO 5000:NEXT W:PAINT(100,100),2
3240 KEY(1) OFF:KEY(2) OFF:KEY(3) OFF:KEY(4) OFF:INTERVAL OFF:GOTO 4340
3250 REM [redacted] VICTORY [redacted]
3260 DIM FLUG%(500):LOCATE30,24:PRINT "VICTORY                              ";
3270 KEY(1) OFF:KEY(2) OFF:KEY(3) OFF:KEY(4) OFF:INTERVAL OFF
3280 PLAY"s0m6000t80","v11t80l8","v11t80l4"
3290 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","n40n52n40n52","n43n55"
3300 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","n39n51v13l32n58n55n51n46n51
n43n46n51v11","n46n58"
3310 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","l8n41n53n41n53","n45n57"
3320 PLAY"l8n24l16n24r32l32n19l16n24n24l32n24l16n19l32n19","n40n52n40n52","n48n6
0"
3330 PLAY"s0m6000t80","v11t80l8","v11t80l4"
3340 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","n40n52n40n52","n43n55"
3350 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","n39n51v13l32n58n55n51n46n51
n43n46n51v11","n46n58"
3360 PLAY"l8n24l16n24r32l32n19l8n24l16n24r32l32n19","l8n41n53n41n53","n45n57"
3370 PLAY"l8n24l16n24r32l32n19t32s13m3000l4n24s0m9000n24","n40n52t32s13m3000l4n4
0s0m9000n40","n48t32s13m3000l4n60s0m9000n60"
3380 LINE(290,160)-(350,180),PSET,7,BF
3390 CIRCLE(320,170),15,2,,,,F
3400 GET@A(290,160)-(350,180),FLUG%,G
3410 FOR Y=160 TO 50 STEP -2
3420  LINE(290,Y+2)-(350,Y+20+2),PRESET,,BF
3430  PUT@A(290,Y)-(350,Y+20),FLUG%,PSET
3440  FOR W=0 TO 200:NEXT W
3450 NEXT Y
3460 GOTO 4340
3470 REM [redacted] TIME COUNTER [redacted]
3480 LOCATE44,8:PRINT TIME$
3490 RETURN
3500 REM [redacted] ｳﾁｱｹﾞ COUNTER (INTERVAL) [redacted]
3510 IF SS=0 THEN GOSUB 3860
3520 IF SS=0 THEN LINE(464,8)-(516,21),PSET,0,BF:LINE(236,9)-(264,69),PSET,0,BF:
LINE(541,6)-(619,24),PSET,0,BF:LINE(480,69)-(OX,OY),PSET,0:GOSUB 3820:INTERVAL O
FF:F=1 ELSE SS=SS-1:LINE(464,8)-(516,21),PSET,0,BF:BEEP:SYMBOL(441,12),STR$(SS),
4,1,7
3530 RETURN
3540 REM [redacted] ﾆﾎﾝﾁｽﾞ DATA [redacted]
3550 M=0:RESTORE3610
3560 READ A$:IF A$="Z" THEN RETURN
3570 IF A$="0" THEN READ CID:M=M+CID-1:GO TO 3600
3580 B=VAL("&H"+A$)
3590 Z%(M)=B+65536!*(B>32767)
3600 M=M+1:GO TO 3560
3610 DATA 0,49,40,0,7,1,D000,0,7,84,0,8,C180,0,7,40,4000,0,7,6018
3620 DATA 0,7,C0,386,3800,0,5,8,4000,3600,0,6,FE0,16,0,6,1E,0,1,188
3630 DATA 0,6,1110,0,1,5000,0,5,53,5E00,3DF0,0,6,590,C061,0,6,2,2010
3640 DATA 4000,0,6,378,740,0,6,1,C800,4000,0,6,4B,0,8,7780,0,7,E7,8000
3650 DATA 0,7,4040,0,7,22,3000,0,7,600C,0,7,10,400,0,7,801,0,7,10,180
3660 DATA 0,7,800,8000,0,6,1C,C0,0,7,1001,8000,0,6,C10,100,0,6,6,7807
3670 DATA 8000,0,4,200,F,C0,400,0,4,5,0,1,600,8002,0,6,7,3F80,4100,0,3
3680 DATA 200,E,0,1,1CE0,1,8000,0,2,2,0,1,18E0,30,0,1,80,0,3,200,3F0
3690 DATA FFC,2000,0,1,C000,0,3,3E,0,1,3F0,0,1,180,0,3,20,2000,0,2,8001
3700 DATA 0,4,1FFF,9FDC,E01C,0,1,280,0,3,218,739,97CF,10C0,809,4000
3710 DATA 0,2,4,1500,8144,C440,A000,A10,0,3,F,C0CF,1C1F,C09C,77A,7000
3720 DATA 0,3,5E0,3306,D80,F7ED,43C0,0,3,1,9011,C98,4080,8C1,0,4,170
3730 DATA 3898,7031,8000,800,0,4,C060,3000,F00,0,5,40,2000,0,2,200,0,4
```

```
3740 DATA 3C20,0,7,3B,E000,0,7,380,0,7,2,0,3,20,0,4,C0,0,7,6,8000,0,12,4000,Z
3750 REM ■■■■ SPST(S) DATA ■■■■
3760 RESTORE 3770:FOR I=0 TO 7:READ S%(I):NEXT I:RETURN
3770 DATA 0,0,0,896,145,-2,32767,-16384
3780 REM ■■■■ MISSILE DATA READ ■■■■
3790 RESTORE 3800:FOR D=0 TO 56:READ M%(D):NEXT D
3800 DATA 1025,16452,4164,1280,24588,384,12294,192,6147,96,3073,-32720,1536,-163
60,768,24588,384,12295,-126,1025,-16260,8135,-513,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,-1,-122,1025,16452,4164,1280,14588,384,12294,192,6147,96,3073,-32720,15
36,-16360,768
3810 DATA 24588,384,12295,-126,0,0,0:RETURN
3820 REM ■■■■ MISSILE -1 ■■■■
3830 MISSILE=MISSILE-1
3840 LINE(313+(7-MISSILE)*15,9)-(323+(7-MISSILE)*15,40),PSET,0,BF
3850 RETURN
3860 REM ■■■■ PSG CONTROLLER ■■■■
3870 GOSUB 4270:SOUND1,0:SOUND3,0:SOUND12,255:SOUND11,255:SOUND6,31:SOUND7,244
3880 SOUND8,15:SOUND9,0
3890 FOR I=200 TO 140 STEP -.2:SOUND0,I:NEXT:SOUND9,15
3900 FOR I=140 TO 40 STEP -.2:SOUND0,I:SOUND2,I+60:NEXT:SOUND8,16:SOUND9,16
3910 SOUND13,0:FOR I=40 TO 10 STEP -.2:SOUND0,I:SOUND2,I+16:SOUND6,I-9:NEXT
3920 RETURN
3930 REM ■■■■ TEMA MUSIC ■■■■
3940 GOSUB 4270'INITIALIZE
3950 SOUND 8,15:SOUND 9,15:SOUND 10,13:'Volume
3960 SOUND 0,146:FOR I=0 TO 3000:NEXT I
3970 SOUND 2,109:FOR I=0 TO 3000:NEXT I
3980 SOUND 4,82 :FOR I=0 TO 3000:NEXT I
3990 SOUND 0,0:SOUND2,0:SOUND 4,0:FOR I=0 TO 500:NEXT I
4000 SOUND 0,146:FOR I=0 TO 100:NEXT I
4010 SOUND 2,109:FOR I=0 TO 100:NEXT I
4020 SOUND 4,82:FOR I=0 TO 1000:NEXT I
4030 SOUND 4,86:FOR I=0 TO 1000:NEXT I
4040 SOUND 4,0:SOUND 8,13:FOR I=0 TO 1000:NEXT I
4050 SOUND 2,130:SOUND 8,15:SOUND 0,146:FOR I=0 TO 500:NEXT I
4060 SOUND 2,97:FOR I=0 TO 2000:NEXT I
4070 FOR I=15 TO 0 STEP -1:SOUND 8,I:SOUND 9,I:SOUND 10,I:FOR W=0 TO 200:NEXT W:
NEXT I
4080 RETURN
4100 REM ■■■■■■■■ RYU-SEI ■■■■■■■■
4110 FOR L=0 TO 10:O=0
4120 RX=300:RY=100:RXINC=RND(1)*2-RND(1)*2:RYINC=RND(1)*.02-RND(1)*.02
4130 WHILE RY<199 AND RY>20 AND RX>0 AND RX<500
4140 PSET(RX,RY,0):O=O+1:RY=RY+RYINC*O^2:RX=RX+RXINC*O:PSET(RX,RY,INT(RND(1)*6+1
))
4150 WEND
4160 NEXT L
4170 RETURN
4180 REM ■■■■ PSG PART 2 ■■■■
4190 SOUND6,5:SOUND7,247:SOUND8,16:SOUND11,0:SOUND12,20:SOUND13,0
4200 RETURN
4210 REM ■■■■ ZUCKON ZUCKON ■■■■
4220 GOSUB 4270:C=1:FOR A=1 TO 5:FORZ=0TO100:NEXT Z
4230 C=C+1:IF C=3 THEN C=1:SOUND 8,15:SOUND 9,15:SOUND 10,15:FOR I=255 TO 70 STE
P -1:SOUND 0,I:NEXT I
4240 COLOR=(1,C):COLOR=(2,C+2):FORCO=3TO7:COLOR=(CO,C):NEXT CO
4250 NEXTA:FOR C=1TO7:COLOR=(C,C):NEXT C
4260 GOSUB 4270:RETURN
4270 REM ■■■■ PSG INITIALIZE ■■■■
4280 SOUND 7,56:FOR SOU=8 TO 10:SOUND SOU,0:NEXT SOU
4290 FOR SOU=0 TO 6:SOUND SOU,0:NEXT SOU:FOR SOU=11 TO 13:SOUND SOU,0:NEXT SOU
4300 RETURN
4310 REM ■■■■ TIME OVER ■■■■
4320 SYMBOL(20,100),"TIME OVER",8,6,2
4330 KEY(1) OFF:KEY(2) OFF:KEY(3) OFF:KEY(4) OFF:INTERVAL OFF:GOTO 4340
4340 REM ■■■■ ONE MORE PLAY ■■■■
4350 LOCATE15,20:PRINT "ONE MORE PLAY (Y/N)":A$=INKEY$
4360 IF A$="Y" THEN CLEAR:MORE=1:GOTO 10
4370 IF A$="N" THEN WIDTH80,25:END
4380 GOTO 4350
4390 END
```

for FM-7

マシン語でPSGをコントロールしよう!

PSGサウンドエディタ

ゲームやシミュレーションなどのプログラムに，効果音は必要不可欠なものです。しかし，マシン語では，BASICと違って，簡単にはPSGをコントロールすることができません。そこで，BASICのSOUNDやPLAY文程度に簡単にPSGをコントロールできるように，サウンドサブルーチンと，そのデータを作成する簡易エディタを作ってみました。

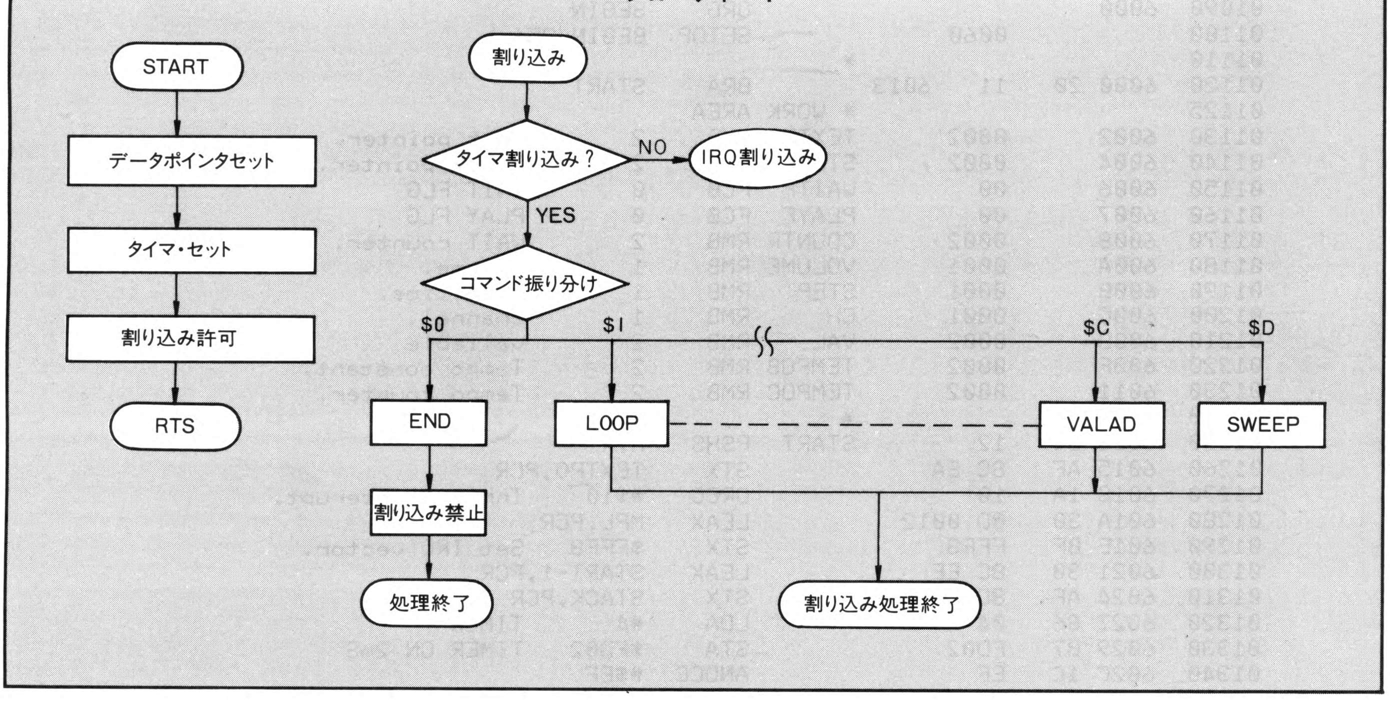
Minky soft　企画部

小　高　康　弘

サウンドサブルーチンの概要

サウンドサブルーチンをここでは，Sound Macro Language (以下SMLと略記) と呼びますが，名前からわかるように，いくつかのマクロ命令を解読しながら実行するインタプリタのようなものです。ユーザープログラムからは，データの先頭アドレスをXレジスタに入れて，SMLをコールし，その際SMLは，FM-7内蔵のインタバルタイマを用いて割り込み処理を行っているので，見かけ上は，ユーザープログラムと同時進行します。このため，ゲームプログラムなどでは複雑な音出しのための手続きが必要なく，非常に簡単に効果音などが出せます。

フローチャート



SMLのプログラム自体は，アセンブルリスト（**リスト1**）および**フローチャート**を見ていただければわかるように，簡単な構成になっています。

最初にユーザープログラムからコールされると，データの先頭アドレスをバッファにストアして，2msのインタバルタイマをイネーブルにし，いったん処理を止めてユーザープログラムに戻ります。そのあと，2msごとに割り込みがかかると少しずつデータをフェッチして処理をし，データが終ると割り込みを禁止してすべての処理を終えます。

データエディタ

データの形式は**表1**に示すとおり，少し複雑です。そこで，比較的簡単にデータを作成できるように，BASICでエディタを作ってみました(**リスト2**)。使い方は実際にプログラムを入力してRUNさせていただければわかると思いますが，最初に動作メニューを聞いてくるので，1～5を入力してください。それぞれの働きは次に示すとおりです。

表1　データの形式

命　令	コード	第1オペランド	第2オペランド	第3オペランド	第4オペランド
END	$00				
LOOP	$01	ループ回数			
LEND	$02				
SOUND	$03	レジスタNo.	データ		
WAIT	$04	WAIT数（2バイト）			
NOTE	$05	チャンネルA~C	音程データ（2バイト）		
NOISE	$06	ノイズ周波数			
ENV	$07	チャンネル	周　期（2バイト）		パターン
VOL	$08	チャンネル	データ $0～$F		
PLAY	$09	チャンネル	音程データ（2バイト）		長　さ
TEMPO	$0A	データ（2バイト）			
VAL	$0B	データ（2バイト）			
VALAD	$0C	データ（2バイト）			
SWEEP	$0D	チャンネル			

1：データの作成
　サウンドデータを作成する場合に選びます。

2：データの読み出し
　1度作成したデータを読み出します。

3：SMLのセーブ
　サウンドサブルーチンを任意のアドレスにセーブします。

4：テスト
　作成したデータを実際に実行します。

5：END
　エディタの実行を終了します。

リスト1　アセンブルリスト（リスト2のデータ部分）

○ここで扱われるプログラムは，個人で利用するほかは著作権法上無断複製を禁じられています。
COPY RIGHT © 1983. Minky Software Systems.

```
                          TTL    Sound Macro Language.
                          OPT    SYMBOL
                  *
                  * Copyright (C) 1983
                  *       by Minky Software Systems.
                  * Programmed
                  *     by T.Kasasaku / Y.Odaka
                  *
            6000  BEGIN   EQU    $6000
6000                      ORG    BEGIN
            0060          SETDP  BEGIN/256
                  *
6000 20     11    6013    BRA    START
                  * WORK  AREA
6002        0002  TEXTPO  RMB    2        Text pointer.
6004        0002  STACK   RMB    2        Stack pointer.
6006        00    WAITF   FCB    0        WAIT FLG
6007        00    PLAYF   FCB    0        PLAY FLG
6008        0002  COUNTR  RMB    2        WAIT counter.
600A        0001  VOLUME  RMB    1        volume.
600B        0001  STEP    RMB    1        step time.
600C        0001  CH      RMB    1        channel.
600D        0002  VAL     RMB    2        valiable
600F        0002  TEMPOB  RMB    2        Tempo constant.
6011        0002  TEMPOC  RMB    2        Tempo counter.
                  *
6013 34     12    START   PSHS   A,X
6015 AF     8C EA         STX    TEXTPO,PCR
6018 1A     10            ORCC   #$10     Inhibit interupt.
601A 30     8D 0012       LEAX   MPL,PCR
601E BF     FFF8          STX    $FFF8    Set IRQ vector.
6021 30     8C EE         LEAX   START-1,PCR
6024 AF     8C DD         STX    STACK,PCR
6027 86     04            LDA    #4       TIMER
6029 B7     FD02          STA    $FD02    TIMER ON 2mS
602C 1C     EF            ANDCC  #$EF
```

```
602E 35    92              PULS  A,X,PC
                      * COMMAND HANDLER             コマンド分析
6030 B6    FD03       MPL  LDA   $FD03
6033 85    04              BITA  #4                 タイマ割り込み?
6035 27    03   603A       BEQ   MPL1               YES
6037 7E    01DD            JMP   $01DD              No。BASICの割り込みルーチンへ
603A 1F    50         MPL1 TFR   PC,D
603C 1F    8B              TFR   A,DP               DPRセット
603E DE    04              LDU   STACK              LOOP~LEND用スタックセット
6040 9E    02              LDX   TEXTPO             データポインタロード
6042 A6    80              LDA   ,X+                コマンドGET
6044 27    0C   6052       BEQ   MPL3               ENDコマンド?
6046 48                    ASLA
6047 31    8D 0101         LEAY  JMPTBL-2,PCR
604B EC    A6              LDD   A,Y
604D 1F    52              TFR   PC,Y
604F AD    AB         MPL2 JSR   D,Y                各コマンドをコール
6051 3B                    RTI
6052 8E    01DD       MPL3 LDX   #$01DD             IRQベクタ復帰
6055 BF    FFF8            STX   $FFF8
6058 7F    FD02            CLR   $FD02              タイマ割り込み禁止
605B 3B                    RTI
                      *
605C A6    80         LOOP LDA   ,X+                ループ回数GET
605E 36    12              PSHU  A,X                ループ回数,くり返しアドレスセーブ
6060 DF    04              STU   STACK              スタックポインタ更新
6062 20    0C   6070       BRA   LEND2
                      *
6064 6A    C4         LEND DEC   ,U                 ループ回数 -1
6066 27    04   606C       BEQ   LEND1
6068 AE    41              LDX   1,U                ループの先頭へジャンプ
606A 20    04   6070       BRA   LEND2
606C 33    43        LEND1 LEAU  3,U                スタックポインタ更新
606E DF    04              STU   STACK
6070 9F    02        LEND2 STX   TEXTPO             データポインタ更新
6072 39              LEND3 RTS
                      *
6073 E6    80        NOISE LDB   ,X+                ノイズ周波数ロード
6075 86    06              LDA   #6             R.6
6077 8D    08   6081       BSR   WRITE              PSGに出力
6079 20    F5   6070       BRA   LEND2
                      *
607B EC    81        SOUND LDD   ,X++               レジスタNo,データGET
607D 8D    02   6081       BSR   WRITE              PSGに出力
607F 20    EF   6070       BRA   LEND2
                      *
6081 34    06        WRITE PSHS  A,B                PSGに出力。A=レジスタNo. B=データ
6083 C6    03              LDB   #3                 ライトコマンド
6085 F7    FD0D            STB   $FD0D
6088 B7    FD0E            STA   $FD0E              レジスタNo.セット
608B 7F    FD0D            CLR   $FD0D
608E E6    61              LDB   1,S                データGET
6090 F7    FD0E            STB   $FD0E              データ出力
6093 C6    02              LDB   #2                 ストローブ
6095 F7    FD0D            STB   $FD0D
6098 7F    FD0D            CLR   $FD0D
609B 35    86              PULS  A,B,PC
                      *
609D 0D    06         WAIT TST   WAITF              WAIT処理中?
609F 26    07   60A8       BNE   WAIT1              YES
60A1 03    06              COM   WAITF              処理中フラグセット
60A3 EC    81              LDD   ,X++               WAIT数GET
60A5 DD    08              STD   COUNTR
60A7 39                    RTS
60A8 109E  08        WAIT1 LDY   COUNTR
60AB 31    3F              LEAY  -1,Y               WAIT数 -1
60AD 109F  08              STY   COUNTR
60B0 26    C0   6072       BNE   LEND3
60B2 30    02              LEAX  2,X
60B4 0F    06              CLR   WAITF              WAIT処理終了
60B6 20    B8   6070 WAIT2 BRA   LEND2
```

```
                           *
60B8 EC    81              NOTE    LDD     ,X++       チャンネル,データ1 GET
60BA 48                            ASLA
60BB 8D    C4    6081              BSR     WRITE      PSGに出力
60BD 4C                            INCA
60BE E6    80                      LDB     ,X+        データ2 GET
60C0 8D    BF    6081              BSR     WRITE      PSGに出力
60C2 20    F2    60B6              BRA     WAIT2
                           *
60C4 A6    80              ENV     LDA     ,X+        チャンネルGET
60C6 8B    08                      ADDA    #8
60C8 C6    10                      LDB     #16
60CA 8D    B5    6081              BSR     WRITE      エンベロープモード指定
60CC 86    0D                      LDA     #13      R.13
60CE E6    80                      LDB     ,X+
60D0 8D    AF    6081              BSR     WRITE      エンベロープパターン指定
60D2 86    0B                      LDA     #11      R.11
60D4 E6    80                      LDB     ,X+
60D6 8D    A9    6081              BSR     WRITE      エンベロープ周期
60D8 4C                            INCA
60D9 E6    80                      LDB     ,X+
60DB 8D    A4    6081              BSR     WRITE
60DD 20    D7    60B6              BRA     WAIT2
                           *
60DF EC    81              VOL     LDD     ,X++       チャンネル,データGET
60E1 D7    0A                      STB     VOLUME
60E3 8B    08                      ADDA    #8
60E5 8D    9A    6081              BSR     WRITE
60E7 20    CD    60B6              BRA     WAIT2
                           *
60E9 A6    80              SWEEP   LDA     ,X+      channel.
60EB 48                            ASLA
60EC D6    0E                      LDB     VAL+1
60EE 8D    91    6081              BSR     WRITE
60F0 4C                            INCA
60F1 D6    0D                      LDB     VAL
60F3 8D    8C    6081              BSR     WRITE
60F5 20    BF    60B6              BRA     WAIT2
                           *
60F7 EC    81              VALSET  LDD     ,X++
60F9 DD    0D                      STD     VAL
60FB 20    B9    60B6              BRA     WAIT2
                           *
60FD EC    81              VALINC  LDD     ,X++
60FF D3    0D                      ADDD    VAL
6101 DD    0D                      STD     VAL
6103 20    B1    60B6              BRA     WAIT2
                           *
6105 EC    81              TEMPO   LDD     ,X++
6107 DD    0F                      STD     TEMPOB
6109 20    AB    60B6      TEMPO1  BRA     WAIT2
                           *
610B 16    FF73  6081      WRITE1  LBRA    WRITE
                           *
610E 0D    07              PLAY    TST     PLAYF      プレイ処理中?
6110 26    1F    6131              BNE     PLAY3      YES
6112 03    07                      COM     PLAYF      プレイ処理フラグセット
6114 EC    81                      LDD     ,X++       チャンネル,データ1 GET
6116 97    0C                      STA     CH         チャンネルストア
6118 48                            ASLA
6119 8D    F0    610B              BSR     WRITE1
611B 4C                            INCA
611C E6    80                      LDB     ,X+
611E 8D    EB    610B              BSR     WRITE1
6120 A6    80                      LDA     ,X+        長さGET
6122 97    0B                      STA     STEP
6124 96    0C                      LDA     CH
6126 8B    08                      ADDA    #8
6128 D6    0A                      LDB     VOLUME
612A 8D    DF    610B              BSR     WRITE1
612C DC    0F              PLAY1   LDD     TEMPOB     テンポセット
```

```
612E DD   11                  STD    TEMPOC
6130 39             PLAY2     RTS
6131 109E 11        PLAY3     LDY    TEMPOC
6134 31   3F                  LEAY   -1,Y
6136 109F 11                  STY    TEMPOC
6139 26   F5   6130           BNE    PLAY2
613B 8D   EF   612C           BSR    PLAY1
613D 0A   0B                  DEC    STEP
613F 26   EF   6130           BNE    PLAY2
6141 96   0C                  LDA    CH
6143 8B   08                  ADDA   #8
6145 5F                       CLRB
6146 8D   C3   610B           BSR    WRITE1   音を止める
6148 0F   07                  CLR    PLAYF    プレイ処理終了
614A 30   04                  LEAX   4,X      データポインタ更新
614C 20   BB   6109           BRA    TEMPO1
                    *
614E      000D      JMPTBL    FDB    LOOP-MPL2
6150      0015                FDB    LEND-MPL2
6152      002C                FDB    SOUND-MPL2
6154      004E                FDB    WAIT-MPL2
6156      0069                FDB    NOTE-MPL2
6158      0024                FDB    NOISE-MPL2
615A      0075                FDB    ENV-MPL2
615C      0090                FDB    VOL-MPL2
615E      00BF                FDB    PLAY-MPL2
6160      00B6                FDB    TEMPO-MPL2
6162      00A8                FDB    VALSET-MPL2
6164      00AE                FDB    VALINC-MPL2
6166      009A                FDB    SWEEP-MPL2
                    *
          0169      SIZE      EQU    *-BEGIN+1
                    *
          6000                END    BEGIN
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=6000
PROGRAM END   ADDR=6167
PROGRAM ENTRY ADDR=6000


BEGIN  6000  CH     600C  COUNTR 6008  ENV    60C4  JMPTBL 614E  LEND   6064
LEND1  606C  LEND2  6070  LEND3  6072  LOOP   605C  MPL    6030  MPL1   603A
MPL2   604F  MPL3   6052  NOISE  6073  NOTE   60B8  PLAY   610E  PLAY1  612C
PLAY2  6130  PLAY3  6131  PLAYF  6007  SIZE   0169  SOUND  607B  STACK  6004
START  6013  STEP   600B  SWEEP  60E9  TEMPO  6105  TEMPO1 6109  TEMPOB 600F
TEMPOC 6011  TEXTPO 6002  VAL    600D  VALINC 60FD  VALSET 60F7  VOL    60DF
VOLUME 600A  WAIT   609D  WAIT1  60A8  WAIT2  60B6  WAITF  6006  WRITE  6081
WRITE1 610B
```

リスト2 エディタプログラム

```
1000 '
1010 ' Sound editor
1020 ' Copyright (C) by ETF Laboratory./Minky soft
1030 ' Programmed by T.Kasasaku / Y.Odaka
1040 '
1050 CLEAR ,&H4FFF:WIDTH 40,20
1060 PRINT "DATA FREE AREA $5000 -- $5F80"
1070 PRINT "PUSH ANY KEY TO START"
1080 DEF FNA$(A$)=RIGHT$(A$,LEN(A$)-1)
1090 GOSUB 2100:GOSUB 2230:GOSUB 2500:GOSUB 2130
1100 A$=INPUT$(1)
1110 COLOR 4:PRINT "*** JOB MENU ***":PRINT
1120 COLOR 7:PRINT "1 ..... DATA ｻｸｾｲ"
1130 PRINT "2 ..... DATA ﾖﾐﾀﾞｼ"
1140 PRINT "3 ..... SML ｽﾄｱ"
1150 PRINT "4 ..... TEST"
1160 PRINT "5 ..... END":PRINT:CLOSE
1170 A$=INPUT$(1):IF A$<"1" OR A$>"5" THEN 1110
1180 ON VAL(A$) GOTO 1610,1200,2180,2150
```

```
1190 END
1200 ' READ DATA
1210 PRINT "OUTPUT DEVICE ? (P/C)":A$=INPUT$(1)
1220 IF A$<>"P" AND A$<>"C" THEN 1210
1230 IF A$="P" THEN OPEN "O",1,"LPT0:":P=1 ELSE OPEN "O",1,"SCRN:":P=0:LN=0
1240 INPUT "START ADDRESS ? = ",A$:AD=VAL("&H"+A$)
1250 IF P=1 THEN INPUT "  END ADDRESS ? = ",A$:E=VAL("&H"+A$)
1260 GOSUB 1340
1270 PRINT #1,ADDR$;" ";NM$;TAB(10);OP$
1280 IF P=1 THEN IF E<AD THEN 1110 ELSE 1260
1290 LN=LN+1:IF LN=15 THEN 1300 ELSE 1260
1300 PRINT "SPACE -- ツヅケル , ESC -- ヤメル"
1310 A$=INPUT$(1):IF A$=" " THEN LN=0:GOTO 1260
1320 IF A$=CHR$(27) THEN 1110 ELSE 1300
1330 ' ONE LINE READ
1340 ADDR$=LEFT$("0000",4-LEN(HEX$(AD)))+HEX$(AD)
1350 IF PEEK(AD)>13 THEN 1370 ELSE NM$=NMO$(PEEK(AD))
1360 ON PEEK(AD)+1 GOTO 1390,1410,1390,1430,1450,1470,1410,1540,1500,1560,1450,1
450,1450,1520
1370 NM$="? ? ?":OP$="":AD=AD+1:RETURN
1380 ' END
1390 OP$="":AD=AD+1:RETURN
1400 ' LOOP & NOISE
1410 OP$=STR$(PEEK(AD+1)):AD=AD+2:RETURN
1420 ' SOUND
1430 OP$=STR$(PEEK(AD+1))+","+FNA$(STR$(PEEK(AD+2))):AD=AD+3:RETURN
1440 ' WAIT , TEMPO , VAL & VALAD
1450 OP$=STR$(PEEK(AD+1)*256+PEEK(AD+2)):AD=AD+3:RETURN
1460 ' NOTE
1470 GOSUB 1600:OP$=OP$+","+FNA$(STR$(INT(76800!/(PEEK(AD+3)*256+PEEK(AD+2)))))
1480 AD=AD+4:RETURN
1490 ' VOL
1500 GOSUB1600:OP$=OP$+","+FNA$(STR$(PEEK(AD+2))):AD=AD+3:RETURN
1510 ' SWEEP
1520 GOSUB 1600:AD=AD+2:RETURN
1530 ' ENV
1540 GOSUB 1600:OP$=OP$+","+FNA$(STR$(PEEK(AD+4)*256+PEEK(AD+3)))+","+FNA$(STR$(
PEEK(AD+2)))
1550 AD=AD+5:RETURN
1560 GOSUB 1600:AA=PEEK(AD+2)+PEEK(AD+3)*256
1570 FOR I=0 TO 96:IF AA=N(I) THEN 1580 ELSE NEXT
1580 OP$=OP$+","+FNA$(STR$(I))+","+FNA$(STR$(INT(192/PEEK(AD+4))))
1590 AD=AD+5:RETURN
1600 OP$=" "+CHR$(PEEK(AD+1)+ASC("A")):RETURN
1610 ' WRITE DATA
1620 CONSOLE 18,2
1630 INPUT "セントウアドレス =",A$:ADR=VAL("&H"+A$)
1640 CLS 1:INPUT "メイレイ ? (E-オワリ,A-アドレス ヘンコウ) =",A$
1650 IF A$="E" THEN WIDTH 40,20:GOTO 1110
1660 IF A$="A" THEN 1630
1670 FOR I=0 TO 13:IF A$=NMO$(I) THEN 1690 ELSE NEXT I
1680 BEEP:GOTO 1640
1690 POKE ADR,I:AD=ADR 'STORE COMMAND
1700 ON I+1 GOSUB 1740,1750,1740,1780,1830,1860,1890,1910,1960,1990,2040,2040,20
40,2070
1710 GOSUB 1330:LI=LI+1:IF LI=17 THEN CLS 2:LI=0
1720 LOCATE 0,LI:PRINT CHR$(5);ADDR$:" ";NM$;TAB(10);OP$:GOTO 1640
1730 ' END & LEND
1740 ADR=ADR+1:RETURN
1750 ' LOOP
1760 INPUT "ループカイスウ (0-255) =",A:IF A<0 OR A>255 THEN 1760
1770 POKE ADR+1,A:ADR=ADR+2:RETURN
1780 ' SOUND
1790 INPUT "レジスタ No. =",R:IF R<0 OR R>13 THEN 1790
1800 INPUT "データ =",D:IF D<0 OR D>255 THEN 1800
1810 POKE ADR+1,R:POKE ADR+2,D:ADR=ADR+3:RETURN
1820 ' WAIT
1830 INPUT "WAIT スウ (0-65535) =",A:IF A<0 OR A>65535! THEN 1830
1840 POKE ADR+1,INT(A/256):POKE ADR+2,A-INT(A/256)*256:ADR=ADR+3:RETURN
1850 ' NOTE
1860 GOSUB 2080:INPUT "シュウハスウ (20-76800Hz) =",A:IF A<0 OR A>76800! THEN 1860
1870 A=76800!/A:POKE ADR+2,A-INT(A/256)*256:POKE ADR+3,INT(A/256):ADR=ADR+4:RETU
RN
```

```
1880 ' NOISE
1890 INPUT "ｼｭｳﾊｽｳ (0-31) =",A:IF A<0 OR A>31 THEN 1890 ELSE 1770
1900 ' ENV
1910 GOSUB 2080:INPUT "ｼｭｳｷ (0-65535) =",A:IF A<0 OR A>65535! THEN 1910
1920 POKE ADR+4,INT(A/256):POKE ADR+3,A-INT(A/256)*256
1930 INPUT "ﾊﾟﾀｰﾝ (0-15) =",A:IF A<0 OR A>15 THEN 1930
1940 POKE ADR+2,A:ADR=ADR+5:RETURN
1950 ' VOL
1960 GOSUB 2080:INPUT "ｵｵｷｻ (0-15) =",A:IF A<0 OR A>15 THEN 1960
1970 POKE ADR+2,A:ADR=ADR+3:RETURN
1980 ' PLAY
1990 GOSUB 2080:INPUT "ｵﾝﾃｲ =",A:IF A<0 OR A>96 THEN 1990
2000 POKE ADR+2,N(A)MOD256:POKE ADR+3,N(A)¥256
2010 INPUT "ﾅｶﾞｻ =",A:IF A<1 OR A>192 THEN 2010
2020 POKE ADR+4,192/A:ADR=ADR+5:RETURN
2030 ' VAL & VALAD
2040 INPUT "ﾃﾞｰﾀ (0-65535) =",A:IF A<0 OR A>65535! THEN 2040
2050 POKE ADR+1,INT(A/256):POKE ADR+2,A-INT(A/256)*256:ADR=ADR+3:RETURN
2060 ' SWEEP
2070 GOSUB 2080:ADR=ADR+2:RETURN
2080 INPUT "ﾁｬﾝﾈﾙ =",A$:IF A$<>"A" AND A$<>"B" AND A$<>"C" THEN 2080
2090 POKE ADR+1,ASC(A$)-ASC("A"):RETURN
2100 DIM NMO$(13):RESTORE 2120
2110 FOR I=0 TO 13:READ A$:NMO$(I)=A$:NEXT:RETURN
2120 DATA END,LOOP,LEND,SOUND,WAIT,NOTE,NOISE,ENV,VOL,PLAY,TEMPO,VAL,VALAD,SWEEP
2130 A#=32.703:DIM N(96):FOR I=1 TO 96
2140 N(I)=INT(76800!/A#):A#=A#*1.0594631#:NEXT:N(0)=0:RETURN
2150 ' DATA TEST
2160 INPUT "DATA ｾﾝﾄｳｱﾄﾞﾚｽ =",A$:A=VAL("&H"+A$)
2170 POKE &H6201,A¥256:POKE &H6202,A MOD 256:EXEC &H6200:GOTO 1110
2180 ' SML SAVE
2190 INPUT "ｾﾝﾄｳｱﾄﾞﾚｽ? =",A$
2200 S=VAL("&H"+A$):IF (S AND 255)<>0 THEN 2190
2210 RESTORE 2270:FOR I=S TO S+&H167
2220 READ A$:POKE I,VAL("&H"+A$):NEXT:GOTO 1110
2230 RESTORE 2270:FOR I=&H6000 TO &H6167
2240 ' SML SAVE
2250 READ A$:POKE I,VAL("&H"+A$)
2260 NEXT:RETURN
2270 DATA 20,11,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2280 DATA 00,00,00,34,12,AF,8C,EA,1A,10,30,8D,00,12,BF,FF
2290 DATA F8,30,8C,EE,AF,8C,DD,86.04,B7,FD,02,1C.EF,35,92
2300 DATA B6,FD,03,85,04.27,03,7E,01,DD,1F,50,1F,8B,DE,04
2310 DATA 9E,02.A6,80,27,0C,48,31,8D,01,01,EC,A6,1F,52,AD
2320 DATA AB,3B,8E,01,DD,BF,FF,F8,7F.FD,02,3B,A6,80,36,12
2330 DATA DF,04,20,0C,6A,C4,27,04,AE,41,20,04,33,43,DF,04
2340 DATA 9F,02,39,E6,80,86,06,8D,08,20,F5,EC,81,8D,02,20
2350 DATA EF,34,06,C6,03,F7,FD,0D,B7,FD,0E,7F,FD,0D,E6,61
2360 DATA F7,FD,0E,C6,02,F7,FD,0D,7F,FD,0D,35,86,0D,06,26
2370 DATA 07,03,06,EC,81,DD,08,39,10,9E,08,31,3F,10,9F,08
2380 DATA 26,C0,30,02,0F,06,20,B8,EC,81,48,8D,C4,4C,E6,80
2390 DATA 8D,BF,20,F2,A6,80,8B,08,C6,10,8D,B5,86,0D,E6,80
2400 DATA 8D,AF,86,0B.E6,80,8D,A9,4C,E6,80,8D,A4,20,D7,EC
2410 DATA 81,D7,0A,8B,08,8D,9A,20,CD,A6,80,48,D6,0E,8D,91
2420 DATA 4C,D6,0D,8D,8C,20,BF,EC,81,DD,0D,20,B9,EC,81,D3
2430 DATA 0D,DD,0D,20,B1,EC,81,DD,0F,20,AB,16,FF,73,0D,07
2440 DATA 26,1F,03,07,EC,81,97,0C,48,8D,F0,4C,E6,80,8D,EB
2450 DATA A6,80,97,0B,96,0C,8B,08,D6,0A,8D,DF,DC,0F,DD,11
2460 DATA 39,10,9E,11,31,3F,10,9F,11,26,F5,8D,EF,0A,0B,26
2470 DATA EF,96,0C,8B,08,5F,8D,C3,0F,07,30,04,20,BB,00,0D
2480 DATA 00,15,00,2C,00,4E,00,69,00,24,00,75,00,90,00,BF
2490 DATA 00,B6,00,A8,00,AE,00,9A
2500 RESTORE 2520:FOR I=&H6200 TO &H6205
2510 READ A$:POKE I,VAL("&H"+A$):NEXT:RETURN
2520 DATA 8E,00,00,7E,60,00
```

マクロ命令(SML)

●END

パラメータ：なし

解説：データ列の最後を示し，サウンドサブルーチンをクローズします。

●LOOP

パラメータ：ループ回数（0～255）

解説：LOOP～LEND間を指定された回数だけ繰り返します。ループ回数が0のときは256回と見なされます。また，LOOP～LENDは，メモリの許す限りネスティングできます。

●LEND

パラメータ：なし

解説：ループの終りを示します。

●SOUND

パラメータ：レジスタNo.，データ

解説：F-BASICのSOUND文と同様です。PSGの各レジスタに直接データを書き込む場合に用います。

●WAIT

パラメータ：ウェイト数（0～65535）

解説：SMLの処理を一時停止します。停止時間はウェイト数1あたり2msです。

●NOTE

パラメータ：チャンネル（A～C），周波数（20～76800Hz）

解説：指定されたチャンネルのオシレータに指定された周波数をセットします。このとき，そのチャンネルがイネーブルとなっていると，その音が聞えます。

●NOISE

パラメータ：ノイズ周波数（0～31）

解説：ノイズの周波数成分を指定します。実際にノイズを出力する場合は，SOUND命令により，ノイズ出力をイネーブルな状態にしておかねばなりません。

●ENV

パラメータ：チャンネル，周期，パターン

解説：指定されたチャンネルをエンベロープモードにし，トリガをかけます。

●VOL

パラメータ：チャンネル，データ(0～15)

解説：指定されたチャンネルの音量をセットします。データが大きくなるほど大きな音になります。

●PLAY

パラメータ：チャンネル，音程，長さ

解説：音楽を演奏します。音程は1～96で指定し，これはF-BASICの，MMLのN1～N96に相当します。長さは四分音符のときは4，八分音符のときは8，というようにします。また，休符は音程＝0とします。

●TEMPO

パラメータ：テンポ

解説：PLAY命令の演奏速度を指定します。

●VAL

パラメータ：データ（0～65535）

解説：SWEEP命令用変数に，データを代入します。

●VALAD

パラメータ：データ（0～65535）

解説：SWEEP命令用変数にデータを加えます。内部は2バイト演算となっているので，65535を加えると1を引いたのと同じです。

●SWEEP

パラメータ：チャンネル

解説：NOTE命令と基本的に変りはありませんが，周波数データとして変数の値を用います。

データサンプル

SMLおよびエディタを用いて，いくつかの効果音を作ってみました（**リスト3**）。それぞれトーンタイプ，ノイズタイプなどありますので，各自，細かなデータ（ループ数，初期データ）などを変えて，いろいろと実行してみてください。それから，実際にどのようなデータをPSGにセットしたらよいかについては触れませんでしたが，F-BASICの文法書，システム仕様にたいへん詳しく書かれているので，参考にしてください。

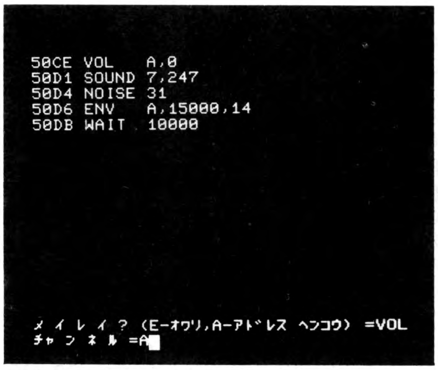


写真1　波のデータ作成

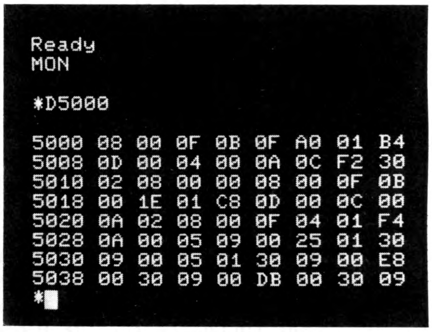


写真2　MONコマンドでメモリに入ったデータをみる

メニュー1で**写真1**のようにデータを作成し，メニュー2で確認，メニュー4でテストします。

リスト3　データサンプル

```
5000 VOL    A,15      チャンネルA，音量15        501C SWEEP A        音出し
5003 VAL    4000      変数←4000代入             501E VALAD 10       変数←変数+10
5006 LOOP   180       180回ループ                5021 LEND           繰り返し
5008 SWEEP  A                                    5022 VOL   A,15     音量15
500A WAIT   10        20ms ウェイト              5025 WAIT  500      1s ウェイト
500D VALAD  62000                                5028 TEMPO 5        テンポ5
5010 LEND             繰り返し                   502B PLAY  A,37,4   ド
5011 VOL    A,0                                  5030 PLAY  A,39,4   レ
5014 VOL    A,15      音量15                     5035 PLAY  A,41,4   ミ
5017 VAL    30        変数←30                   503A PLAY  A,42,4   ファ
501A LOOP   200       200回ループ                503F PLAY  A,44,4   ソ

(FIRE: 5000～5021 / RANDOM NOTE: 5011～5022)
```

```
5044 PLAY  A,46,4     ラ            ┐
5049 PLAY  A,48,4     シ            │
504E PLAY  A,49,4     ド            │
5053 WAIT  500        1s ウェイト    │
5056 PLAY  A,51,4     レ            │ MELODY
505B PLAY  A,53,4     ミ            │
5060 PLAY  A,49,4     ド            │
5065 PLAY  A,37,4     ド            │
506A PLAY  A,44,4     ソ            │
506F WAIT  500        1s ウェイト    ┘
5072 SOUND 7,246      ノイズ イネーブル Ach ┐
5075 VOL   A,15       音量15        │
5078 LOOP  30         30回ループ     │
507A NOISE 15         ノイズ周波数15  │ FLY
507C WAIT  30         60ms ウェイト  │
507F NOISE 30         ノイズ周波数30  │
5081 WAIT  30         60ms ウェイト  │
5084 LEND             繰り返し       ┘
5085 SOUND 7,248      サウンドイネーブル ┐
5088 VOL   A,0                      │
508B LOOP  50         50回ループ     │
508D VOL   A,10       音量10        │
5090 NOTE  A,200      200Hz出力     │ ???
5094 WAIT  30         ウェイト       │
5097 NOTE  A,100      100Hz出力     │
509B WAIT  50         ウェイト       │
509E LEND             繰り返し       ┘
509F VOL   A,15       音量15        ┐
50A2 VAL   1000       変数←1000     │
50A5 LOOP  200        200回ループ  ┐音が     │
50A7 SWEEP A                      │だんだん  │
50A9 VALAD 65533      変数←変数-3 ┘高くなる  │
50AC LEND             繰り返し       │
50AD VAL   1000  ┐                  │
50B0 LOOP  200   │                  │
50B2 SWEEP A     │                  │
50B4 VALAD 65532 │                  │ SWEEP
50B7 LEND        │                  │
50B8 VAL   1000  │  上記の繰り返し   │
50BB LOOP  200   │                  │
50BD SWEEP A     │                  │
50BF VALAD 65532 │                  │
50C2 LEND        │                  │
50C3 VAL   1000  │                  │
50C6 LOOP  0     │                  │
50C8 SWEEP A     │                  │
50CA VALAD 65530 │                  │
50CD LEND        ┘                  ┘
50CE VOL   A,0                      ┐
50D1 SOUND 7,247         ノイズイネーブル │
50D4 NOISE 31            ノイズ周波数31  │ WAVE
50D6 ENV   A,15000,14    エンベロープモード設定(三角波) │
50DB WAIT  10000         ウェイト       │
50DE VOL   A,0           音を止める     ┘
50E1 SOUND 7,248         サウンドイネーブル ┐
50E4 LOOP  50            50回鳴らす     │
50E6 NOTE  A,1505        1505Hzの音     │
50EA WAIT  10            ウェイト       │
50ED VOL   A,15          音量15         │ ALARM
50F0 WAIT  10            ウェイト       │
50F3 VOL   A,0           音を止める     │
50F6 LEND                繰り返し       ┘
50F7 END                 データ終了
```


勉強時間が楽しくなった!!

EXPERT MATHEMATICS

中学数学シリーズ 全36巻

いよいよ本格的に発売!

(FM-7・8・PC-6001・PC-8001・MZ-700・1200)

学年	タイトル	内容	価格	コードNo.
中1	整数 1	約数・倍数・累乗・素因数分解	3,000円	A 1
中1	整数 2	最大公約数・最小公倍数	3,000円	A 2
中1	正の数・負の数	数直線・加減法・乗除法	3,000円	A 3
中1	一次方程式1	等式の性質・移項・一次方程式の解法	3,000円	A 5
中2	式の計算	式の加減・式の乗除・混合問題・式の値	3,000円	B 1
中2	連立方程式1	加減法・代入法、( )の問題、分数の問題	3,000円	B 4
中3	式の計算 1	式の展開・乗法公式の活用	3,000円	C 1
中3	式の計算 2	因数分解・文章応用の問題	3,000円	C 2

EXPERT ENGLISH

中学英作文シリーズ

実績が光る英語、グレードUPして新発売!

(FM-7・FM-8・PC-6001・PC-8001)

学年	Part	第1巻	第2巻	装丁	価格	コードNo.
中1	1	Be動詞・What・Who	have文・命令形	BOOK型2巻セット	5,800円	X12
中1	2	I.You文・複数形・Which	一般動詞・When.How	BOOK型2巻セット	5,800円	X34
中1	3	進行形・There・Can	問題コース付1年英単語辞典	BOOK型2巻セット	5,800円	X50
中2	1	過去形	比較形	BOOK型2巻セット	5,800円	Y12
中2	2	未来形	不定詞・受動態	BOOK型2巻セット	5,800円	Y34
中2	3	2年総復習	問題コース付2年英単語辞典	BOOK型2巻セット	5,800円	Y50
中3	1	動名詞	現在完了形	BOOK型2巻セット	5,800円	Z12
中3	2	現在分詞・過去分詞	関係代名詞	BOOK型2巻セット	5,800円	Z34
中3	3	中学総復習	問題コース付3年英単語辞典	BOOK型2巻セット	5,800円	Z50

※送料は一切必要ありません　※学習ソフトオールカタログ切手150円分同封

好評発売中!

EXPERT HISTORY・GEOGRAPHY

中学社会科シリーズ

日本史・世界史 (PC-6001・PC-8001 FM-7・8・MZ-700・1200)

日本地理・世界地理 各3,000円

高校受験英語シリーズ全20巻(PC-6001)

RERFECT MATHEMATICS

高校数学シリーズ 全50巻 一部発売中

(PC-6001, PC-8001MKII, FM-7・8, パソピア)

SKJ 数研塾

〒411 静岡県三島市一番町15の32 芹沢ビル4F

☎0559-75-1534(三島駅前)

おもしろ★パソコン用語集

数あるコンピュータ関連書籍の中で，読まれない本の代表は用語集である。

たとえば，せっかく2,000円出して買い込んでも20項目しか読まなかったなら，そのコストは1項目あたり100円にもなり，非常に不経済だ。だが，その理由はと考えてみると，どうも用語集側に責任があるようだ。つまり無味乾燥な活字のら列が，読者をあきさせるのだ。

そこで，oh！FMを読むための必要最少限の用語を，楽しく読めるようにまとめてみた。

——————**おもしろくなければ用語集ではない**——————

【RS-232C】直列2進データ転送に関するデータ端末装置と通信装置との接続条件を規定した国際的な標準規約。本来は端末とモデムをつなぐ際の規格として出発したものであり，現在のように「電線による遠距離通信」に用いるのは，多少邪道の感がしないでもない。しかし今や直列のインタフェース規格として，最もポピュラである。

【アイ・シー(IC)】集積回路（Integrated Circuits）のこと。ANDやORなどの論理回路をワンチップの中に多数組み込んでいる。集積度をより大きくしたものはLSIと呼ばれる。You see？

【I／O】①雑誌名。②InputとOutputの略。I／Oカード，IO装置などのように用いる。

【アスキー】①雑誌名。②米国標準協会の制定した情報交換用コード(American Standard Code for Information Interchange)。7ビットの文字セットと1ビットのパリティビットよりなる。日本のJISコードは，このASCIIコードにカナを追加したもの。

【アセンブラ】アセンブリ言語で書かれたプログラムをアセンブルするもの（説明になっていない)。ニモニックという数個のアルファベット文字よりなる記号を機械語に変換するもの。

【アドレス】主記憶に関して付けられた特定の番地。通常8ビット系パソコンでは，アドレスは0から65535（16進数で0000〜FFFF)。

【インストラクション】コンピュータの実行できる最少単位の命令。たとえば「Store Register」はインストラクションネームである。

【インタフェース】2つの物の間で情報をやりとりすること。またはやりとりを行うメカニズム。

【インタプリタ】BASICなどのように，高級言語で書かれたプログラムを解釈しながら，実行するもの。コンパイラより操作が簡単だが処理が遅い。

【インタラプト】割り込みのこと。実行中のプログラムを強制的に中断させ，特別に用意されたサブルーチンへジャンプさせる。人間社会でこれをやると，時に袋だたきにあう。

【ウォームスタート】①ミミズの競走。②コールドスタートを行い，コンピュータを動作させていて，何かの原因でプログラムがコントロールを失ったときに，必要な部分を初期設定して再び正常に戻すこと。

【FM】①東京，NHKなど。静岡も開局した。Frequency Modulation。②富士通のパソコン Fujitsu Micro 7, 8, 11

【エントリポイント】マシン語サブルーチン中の実行開始メモリ番地。通常BASICでは様々なエントリポイントが存在するが，CP/Mのシステムコールでは5番地ただ1個である。

【カタタン】①固い痰。②片面単密度フロッピのこと。同様にリョウバイは両面倍密度のことを言う。

【カラケット】フォーマット処理もなにもされていないディスケットのこと。

これはカラオケ

【逆アセンブラ】ディスアセンブラとも言う。マシン語をアセンブリのニモニックに変換する。ちょうどアセンブラの逆の働きをする。逆噴射はこれとは全く関係ない。

【キャラクタジェネレータ】キャラクタ（文字や記号)のパターンを書き込んであるROMのこと。システムによっては主記憶上にパターンが置かれることもある。

【高級言語】機械語や，アセンブリに対するBASICやFORTRANなどの総称。アセンブリに比べると，一般的にプログラミングやデバッグが容易だが，スピードが遅く，小まわりがきかない。

なお高級というのは機械語などに対して，人間の日常言語に近い度合が高いということであり，これを用いる人が高級というわけではない。

【コントロールキャラクタ】 アスキコードのうち0～31までのものを言う。この32コードは文字を出力するのではなく,PRINT文にて各種のコントロールを行わせるために用いられる。

【コンピュータ】 今さら……。

【コールドスタート】 パワーオフになっている（当然冷えているだろう）コンピュータを，全ての初期設定からスタートさせること。

【CRT (Cathode Ray Tube)】 テレビの画面。またそのような画面を持つ装置。

【スクロール】 画面に表示されているテキストを，上や下に動かすこと，またそのような機能。

【スタック】 情報を一時的に貯えるためのメモリ内の予約された領域。たとえばGOSUB文を実行すると，戻り番地がスタックされる。

【スパゲッティ】 ①食べ物。②GOTO文が多く，やたらに論理のもつれたプログラム（ナーンモわからん)。なお，スパゲッティはアサリが一番うまい。

【チェックサム】 機械語のリストにおいて，入力間違いを防ぐために，番地と機械語を足したもの。人によりチェックサムの計算法に差があるが，Oh！FMでは次のように統一している。

チェックサム＝×××0～×××7または，×××8～×××Fの8バイトの機械語を足したものの下2ケタ。

ブッチャーのサミングはレフェリーがチェックする（それがどうした)。

【テキスト】 もともと文字や数のキャラクタのことであるが，コンピュータでは特に，プログラムリスト（としての文字の集合）をテキストということが多い。

【デフォルト】 ①……のようなもの。②省略時解釈。たとえばAUTO↵とキーインすると，AUTO10，10↵と入力されたものとみなす。このとき10という数値はデフォルト値と呼ばれる。

【トバス】 ①東京都のバス。②アクセルを踏む。③ディスケットを破壊させてしまうこと。あ，トバしちゃった，という具合に使う。

【ドス】 ①凶器。②京都弁。③Disk Operating Systemの略で,ディスクとデータ交換するための基本手順をサポートするシステムプログラム。昔はDisk Oriented systemと言ったのだが，いつから変わったのだろう。なお専門家レベルではDOSとはディスクパックを用いるシステムのことであり，フロッピベースのものはFDOSと呼ばれる。

【ナイコン】 コンピュータを持っていない人たちのこと。同様にニコンは2つ持っている人のこと，ではない。これは日本光学。

【ナミヒコ】 小生のセガレの名前。(カキタカッタノ。ゴメン)

【ニモニック】 覚えるのが難しい2進数の機械語の代わりに，記憶しやすいように作られた頭文字などでできた記号。たとえばLDAやSTAなど。

【ネスティング】 あるサブルーチンから，さらに別のサブルーチンを呼ぶような構造をいう。単にネストということもある。

【ハードウェア】 ①鉄仮面やよろい。②コンピュータの本体や，プリンタなど実体のあるものの総称。

【バイナリ(binary＝2進法)】 0と1の2つの数字で表す数え方。たとえば0001は1，0010は2，0100は4を表す。このように数字1が一つ左に移動するたびに数値が倍になるので「倍なり」と言う(冗句です)。

【バグ】 通常はプログラミング上の欠陥をいう。虫。バグをとることをデバッグという。本誌の訂正欄「虫つくろい」はこれから命名。とった虫を手につまみあげるとハンドバッグになる。

このカットは第2号を
お持ちの方しか知らない。

【バグる】 バグ（虫）が出ていることを言う。他にエラーが出ているときは，エラッてると言ったりする。

【バッファ】 データを一時的に保持させる記憶装置や領域。動作速度の異なる回路(たとえばCPUとプリンタ)間のインタフェースを正しく行わせるために，中間に設けられる。

【パレット機能】 画面の色指定を実際の色と対応させる機能。COLOR 7は通常白を表すが，COLOR＝7，1を実行すると，以後COLOR 7は青を意味することになる。

【BIOS】 バイオスと読む。CRTへの文字表示や，キーボードからの入力など，基本的な入出力機能をまとめたもの。

【ビット】【バイト】 ①どちらも雑誌名。②コンピュータの持っている最小単位の情報をビットという。一つのビットは，0または1の二つの状態を示す。また，通常8ビットまとめたものを1バイトと呼ぶ。したがって雑誌もBYTEの方がbit誌より厚い。

【ファイル】 ある目的のために集められ，組織化された情報の集まり。データファイルとプログラムファイルに大別される。

【ブランク】 のこと。

【プログラマ】 ①グラマーな体を道具に，お金を稼ぐ人。②プログラムを作る人。ただし，実力のない人はアマグラマ。

【プログラム】 処理を記述する連続した命令。プログラムに問題があるとプロブレムになる。

【ベンチマークテスト】 ①ペンキ塗りたての試験。②コンピュータの性能を評価するため，小規模な基本プログラムを実行させ，実行時間などを判断する方法。

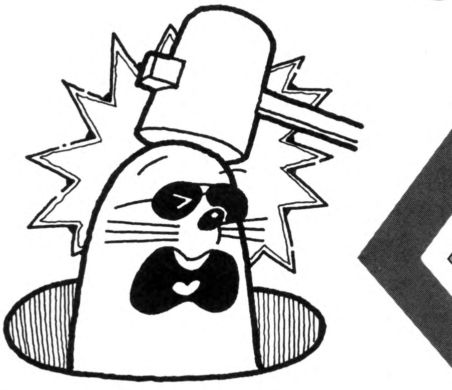
【マルチページ】 画面表示用のメモリを複数組もっていること。ページを切り換えることにより，高速に画面変換を行わせることができる。

【モニタ】 ①実行結果を確認すること。また，するもの(CRTなど)。②コンピュータを非常に低いレベルで走らせるためのプログラム。パソコンでは通常機械語モニタのことを言うが，本来コンピュータにおけるモニタとは，もっとレベルの高いものであり，時にスーパーバイザと呼ばれたりする。

【ラム】 ①アグネス（グラマーだった)。②Random Access Memory。本来，磁気テープなどのシーケンシャルリードの記憶媒体に対して，どの番地でも任意に読み出しができるメモリを意味したが,現在ではROMに対応する用語としてリード／ライトともに可能なメモリという意味で用いられている。

【ロム(Read Only Memory)】 読み出し専用のメモリ。キャラクタパターンや，漢字パターン，あるいはBASICインタプリタの本体を焼き込んでおくと，電源ONですぐ使える。PROM(Programable ROM)や，EPROM(Erasable PROM)などのタイプがある。

自作プログラムを作ろう!



画面制御から入る F-BASIC入門

●石原 宏明

F-BASICの文法書はひととおり読んだ，各コマンドの意味も大体わかった，でも実際にプログラムをどう組んでいいかわからない，という人のためのF-BASIC講座です。決して安くはないパソコンを買い込んだのですから，BASICのプログラムの一つや二つ，自分で作ってみたいと思いませんか。

巷に氾濫するBASIC講座，パソコンスクールなるもの，あまりに画一的でかつわかりにくく，おもしろくないと思いませんか。PRINT文やINPUT文，GOTO文などから入るのも一つの方法かもしれませんが，テレタイプの時代と違い，今やカラーCRTの時代。画面を自由自在にあやつり，好きな所に好きな絵を描くことから入るのが自然でおもしろいと思います。たかだか640×200ドットで「コンピュータグラフィックス」と言うのもおこがましいかもしれませんが，グラフィックからのBASIC入門にはFMは適当なマシンです。そこで私なりに知恵を絞って，巷のパソコンスクールとは一味違ったプログラミングテクニックの講座をお届けします。題して「画面制御から入るF-BASIC入門」です。当面の目標は，もぐらたたきゲームの作成です。

§画面初期設定

(CONSOLE, WIDTH, COLOR, CLS文)

一番はじめに画面初期設定に関するコマンドを説明しましょう。よくご存知の人は，この章を読み飛ばしてください。

画面をどんな状態で使うか，つまり，横文字数・縦行数，スクロール範囲・ファンクションキーの内容表示の有無・単色表示か否かを決定するのが，CONSOLE，WIDTH文です。次のプログラムを入力して，RUNしてみましょう。

```
10 WIDTH 40,25:CONSOLE 5,10,0,0:COLOR 7,0:CLS
20 LINE(0,0)-(639,39),PSET,2,B
30 LOCATE 0,0:PRINT"PAGE 1"
40 LINE(0,120)-(639,199),PSET,4,B
50 LOCATE 0,15:PRINT"PAGE 2"
60 LOCATE 0,5:PRINT"SCROLL"
```

図1

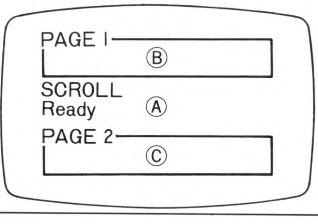


図1のような画面が現われます。PAGE 1と書かれて赤いワクで囲まれた範囲が「ページモード1画面」と呼ばれる部分（Ⓑの部分），PAGE 2と書かれて緑のワクで囲まれた範囲が「ページモード2画面」と呼ばれる部分（Ⓒ），また，SCROLL，Readyと書かれてカーソルが点滅しているのが「スクロール画面」と呼ばれる我々に最もなじみの深い部分（Ⓐ）です。この状態で上下のカーソルキーを使ってカーソルを動かしてください。カーソルはⒷやⒸの範囲には絶対に行かず，あくまでⒶの中を移動します。また，CLEARキーを押してもⒶの部分しか消去されません。LISTをとると，Ⓐの範囲内でのみスクロール*1が行われます。このようにCONSOLE文は画面を最高三つの部分に分けることができます。

```
CONSOLE 5,10,0,0
```

画面Ⓐの開始行 ↑　↑ 画面Ⓐの行数

次にⒷの部分はどうなっているか見てみましょう。次の文をダイレクト実行*2してください。

```
LOCATE 0,0
```

カーソルがⒷの画面に来たでしょう。カーソル移動キーを使ってもⒷの範囲外に移動しないのは，Ⓐのときと同じです。さて，LISTをとってみましょう。プログラムは6行ですから，一度に全部を表示することは不可能です。そこでこの画面ではⒶのようなスクロールは行われず，最下行が表示されるとすぐにⒷの画面全体が消去され，再びⒷの画面左上から次の表示が開始されます。このように，あたかも本のページをめくるようなので「ページモード」と呼びます。このことはⒸの画面についても同じです。

CLEARキーは，現在カーソルのある画面しか消してくれません。

HOMEキーも，現在カーソルのある画面のホーム位置にカーソルを移動させます。

CLS文は，その直後の引数により，Ⓐ，Ⓑ，Ⓒのどの画面を消去するかを決めることができます。

CLSまたはCLS0……Ⓐ，Ⓑ，Ⓒすべて消去。カーソルをⒶのホーム位置へ。
CLS 1……Ⓐを消去。カーソルをⒶのホーム位置へ。
CLS 2……Ⓑを消去。カーソルをⒷのホーム位置へ。
CLS 3……Ⓒを消去。カーソルをⒸのホーム位置へ。

ゲームなどで常にコメントを必要とするときは，

```
WIDTH 40,25:CONSOLE 24,1,0,0
```

とすれば画面最下行1行だけがスクロール画面となり，0～23行はページモード1になりますから，24行目に常にコメントを表示することにすれば，それより上の表示を損なうことがなくなります。ただし，スクロール画面でもページモード画面でも最下行の表示には注意することが一つあります。それはPRINT文は黙っていれば必ず改行を伴うことです。よって最下行を表示したとたん，スクロールまたはページ替えが行われて，せっかくの表示をこわしてしまうことがあります（24行目をコメントライン専用に使うときは，逆にそれを利用するのですが)。このことについては後でもう一度ふれます。

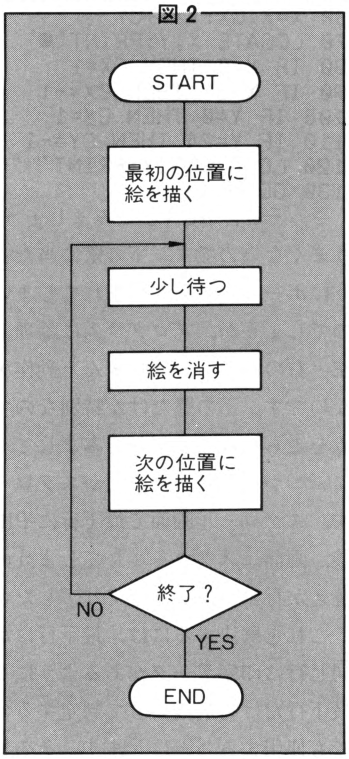
画面初期設定に使うWIDTH，CONSOLE，COLOR，CLS文はプログラムの先頭に置くと見通しがよくなります。

§ボールを動かす

(LOCATE, PRINT, IF～THEN文)

画面上で線を動かすときの基本は，フローチャートに示すと**図2**のようになります。これはBASICでもマシン語でも変わりありませんが，BASICでは絵を描くスピードそのものが遅い関係から「少し待つ」ことができないこともあります。この場合は描いた絵をすぐ消すことになるので，表示がチラチラするのを除くことができません。また，絵を消す仕事が終わったら，間にあれこれと作業をはさまず，すぐに次の絵を表示するようにすれば少しでもよい画質を得ることができます。

図2



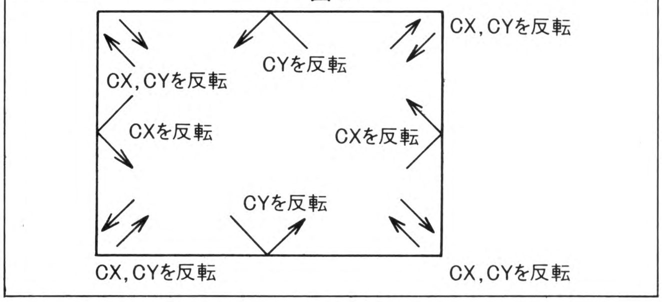
では，壁に当たったボールがはね返るシミュレーション（＝模擬実験）をやってみましょう。

X，Yをボールの横座標，縦座標とし，CX，CYをボールの横方向および縦方向の運動成分とします。CX，CYは－1または1のどちらかの値のみをとるようにして，(X，Y）の位置にボールを表示したら，

```
X=X+CX:Y=Y+CY
```

として，次の（X，Y）の位置にボールを表示すればボールは動いていきます。もちろん，次の表示を行う前に先ほどのボールの位置には空白“␣”を書いてボールを消しておきます。

図3



ボールが壁にぶつかったときは，当然，進行方向が変わりますから，このことも考慮しなければなりません。左右の壁にぶつかったときは横方向の運動成分(CX)を反転し，上下の壁にぶつかったときは縦方向の運動成分(CY)を反転します(**図3**参照)。左右の壁にぶつかったかどうかの判定は，左の壁ならX＝0，右の壁ならX＝39かどうかでできます。同じように上下の壁にぶつかったかどうかは，上の壁ならY＝0，下の壁ならY＝24かどうかでできます。これらをコーディングすると，以下のようになります。

```
IF X=0 THEN CX=1
IF X=39 THEN CX=-1
IF Y=0 THEN CY=1
IF Y=24 THEN CY=-1
```

それでは，いよいよ全体をコーディングしてみましょう。

---

*1 画面がいっぱいになると，巻物のように，新しい行が下にはいるたびにその分一番上の行が画面から消えていく状態。

*2 行番号を書かずに，直接入力したあと⏎キーを押すこと。

```
10 WIDTH 40,25:CONSOLE 0,25,0,0:COLOR 7,0:CLS
20 LOCATE 39,12:PRINT"●"
30 LOCATE 39,12:PRINT" "
40 X=39:Y=12
50 CX=-1:CY=-1
60 X=X+CX:Y=Y+CY
70 LOCATE X,Y:PRINT"●"
80 IF X=0 THEN CX=1
90 IF X=39 THEN CX=-1
100 IF Y=0 THEN CY=1
110 IF Y=24 THEN CY=-1
120 LOCATE X,Y:PRINT" "
130 GOTO 60
```

さっそく，RUNしてみましょう。ところがこのプログラム，うまくないのです。下の壁に当たるたびに，まるで分身の術のようにボールの兄弟が生まれてしまいます。この原因はどこにあるのでしょうか。プログラムに論理上の誤りはなさそうです。上・左・右の壁のときはちゃんと動作するのに，下の壁の時だけだめなのです。下の壁だけが特別なのでしょうか。しばらくプログラムを走らせたままにしてみましょう。どうやら，ボールが下の壁にぶつかったときは画面がスクロールしているようです。そうです，スクロール画面で最下行にPRINT文を黙って使うと，印字後，画面はスクロールしてしまうのです（ページ画面ではページ替えが行われ，画面をクリアしたのと同じ結果になってしまう)。

これを解決するには，最下行にボールを表示した後でも，まだ同じ行に印字データがあるようにBASICに思わせることです。最下行だけ特別扱いして判定するのは面倒なので，どの行においても使用することにします。その具体的な方法とは，行番号70と120の最後にセミコロン「；」を付けることです。

```
70 LOCATE X,Y:PRINT"●";
120 LOCATE X,Y:PRINT" ";
```

これで，最下行に“●”や“␣”が印字されても改行を行いません。しかし，最下行の一番最後，つまり画面の右下の端にボールが来たときには，やはり改行が行われてしまいます。これは，画面をいっぱいにつかっているためで，これを防ぐには，行番号90のIF X＝39をIF X＝38とするか，行番号110のIF Y＝24をIF Y＝23とするなどが一般的です。行番号110を直した場合は，行番号70，120にセミコロンを付ける必要はありません。

さて，以上のことはテニスゲームやブロックくずしを作る際の基本となるものですが，残念ながらBASICではスピードが遅すぎます。先のボール移動プログラムは，ボールをキャラクタ単位で動かすプログラムとしては最小かつ最高速であると考えてもらっていいでしょう。したがって，これ以外の作業が途中に入れば当然スピードが落ちてしまいます。

§絵を動かす

(DIM, GET@, PUT@文)

PRINT文でグラフィックキャラクタを使っていくら工夫した絵でも，LINE，CONNECT，CIRCLE，PSET，PAINTなどのグラフィックステートメントを使って描いた絵にはかないません。その絵をそっくりそのまま他の場所に移動できるのが，GET@，PUT@文です。ここでは，マニュアルのGET@，PUT@文の形式4について説明します。

GET@文の形式4とは，画面に描いた絵をそっくりそのまま写し取ってしまうという便利な命令です。形式3と違って色の情報までが読み取られます。読み取る範囲は，やはりコンピュータですから任意の形でというわけにはいかず，指定した長方形の中のドットデータの情報を読み取ります。GET@，PUT@はF-BASICの命令群の中でも実際に使ってみるまでは何が起こるのかわからないという，最も理解しにくい命令です。イメージとしては，画面上の一部分を写真に撮るようなものと思ってください。写真では色情報はフィルムに焼き付けられますが，GET文では画面上の色情報は配列にしまわれます。

GET＠文によって得られた情報をはき出して画面上に再現するのがPUT＠文です。このようにGET＠とPUT＠は対にして使うのが普通です。また，GET＠で読み込んだ情報を他の位置にそのまま再現するのが最も普通の使い方ですが，PUT＠文をいろいろといじることによって，いろいろな表示の仕方ができます。

まず，ノーマルな使い方をお目にかけましょう。次のプログラムを実行してみてください。

```
10 CLS
20 DIM M%(900)
30 ´*** ｱﾅ ***
40 CIRCLE(100,50),60,7,.15
50 ´*** ﾓｸﾞﾗ ***
60 CIRCLE(100,50),40,6;.6,.45,.05,F:´ﾄﾞｳﾀｲ
70 PAINT(100,51),6,7
80 CIRCLE(115,40),9,7,.6,,,F:´ﾒ
90 CIRCLE(85,40),9,7,.6,,,F
100 PSET(114,36,0):´ﾒﾀﾞﾏ
110 PSET(115,36,0)
120 PSET(116,36,0)
130 PSET(114,37,0)
140 PSET(115,37,0)
150 PSET(116,37,0)
160 PSET(84,42,0)
170 PSET(85,42,0)
180 PSET(86,42,0)
190 PSET(84,43,0)
200 PSET(85,43,0)
210 PSET(86,43,0)
220 CIRCLE(115,40),12,0,.6,.7,.9:´ﾏﾕ
230 CIRCLE(85,40),12,0,.6,.6,.8
240 LINE(67,48)-(133,48),PSET,1:´ﾋｹﾞ
250 LINE(66,45)-(131,51),PSET,1
260 LINE(69,51)-(131,45),PSET,1
270 CIRCLE(90,48),12,0,.4,0,.3:´ｸﾁ
280 CIRCLE(110,48),12,0,.4,.2,.5
290 LINE(95,50)-(105,54),PSET,0,B:´ﾊ
300 LINE(100,50)-(105,54),PSET,0
310 PAINT(98,52),7,0
320 PAINT(102,52),7,0
330 CIRCLE(100,48),9,2,.3,,,F:´ﾊﾅ
340 ´*** ﾓｸﾞﾗ ｦ ﾊｲﾚﾂ ﾆ ﾖﾐｺﾑ ***
350 GET@A (40,27)-(160,58),M%,G
360 ´*** ﾓｸﾞﾗ ｦ ﾍﾞﾂ ﾉ ﾄｺﾛ ﾍ ﾋｮｳｼﾞ ***
370 PUT@A (240,27)-(360,58),M%,PSET
```

もぐらに見えたでしょうか。行番号20でもぐらの色情報を取り込むための整数配列を用意し，行番号30〜330で穴ともぐらを描きます。行番号350がもぐらの絵を配列に取り込むためのGET＠文形式4です。このままでは何も起こりませんが，行番号370で先にGET＠文で取り込んだ絵を別の位置にプリントアウトしま

す。このようにGET@，PUT@文を使えば，いったん描いた絵を瞬時に，そして一挙に別の所へ移すことができます。

配列の大きさの決め方はマニュアルに詳しく書かれていますが，表示範囲のドット数を計算するのは非人間的な作業なので，一般に少し大きく取っておけばいいでしょう。

なお，PUT@文の表示の仕方は，左から右へ，上から下へです。また，B，R，Gの三原色をこの順番で重ね合わせていきます。したがって，大きな絵になるほど，一挙に表示というわけにはいかなくなります。

PUT@文は必ずしもGET@文と対にして使う必要はなく，何んらかの形で色情報が配列に入っていれば，GET@文なしに，PUT@文を使うことができます。

また，行番号20，350，370を次のように変更してみてください。

```
20 DIM M%(1500)
350 GET@A (0,0)-(158,49),M%,G
370 PUT@A (0,100)-(158,149),M%,PSET
```

このように，GET@とPUT@の指定する長方形の形を変えてみると，表示がずれて思わぬ色の絵が表示できます。GET@，PUT@の一風変わった使い方です。

§中間色

(COLOR, LINE, CHR$(254))

FM-8，7は640×200ドット，1ドットごとに8色の色指定といった優れたグラフィック機能を持っています。そこで基本8色だけでなく，色の組み合わせによって中間色を使ってみたくなります。ところが，PC-8801などのようにタイリング機能を持っていないFM-8，7は中間色表示がちょっと苦手なのです。それでもなんとか中間色を使ってみたいということで，割合と簡単に中間色を得る方法を紹介しましょう。それは，CHR$(254)を使う方法です。CHR$(254)とは▓のことです。このCHR$(254)と背景色の組み合わせで，結構いろいろな色を出すことができます。しかし，PRINT文を使う関係で色付けはキャラクタ単位となりますし，背景色との組み合わせでないとうまくありません。また，PAINT文は思ったように働いてくれません。さて，どんな中間色が得られるか，まずは**リスト1**を実行して2色の組み合わせによる中間色をごらんください。

リスト1

○ここで扱われるプログラムは，個人で利用するほかは著作権法上無断複製を禁じられています。
COPY RIGHT © 1983 H. ISHIHARA

```
10 WIDTH 40,25:COLOR 7,0:CLS
20 DIM A%(160),B%(150)
30 FOR I=1 TO 10
40    LOCATE 0,I-1
50    PRINT CHR$(254);CHR$(254);
60 NEXT
70 GET@ (0,0)-(31,79),A%,G
80 GET@ (0,0)-(29,79),B%,G
90 CLS
100 LINE(0,0)-(11,9),CHR$(254),1,BF
110 LINE(12,0)-(21,9),CHR$(254),2,BF
120 LINE(22,0)-(29,9),CHR$(254),3,BF
130 LINE(30,0)-(35,9),CHR$(254),4,BF
140 LINE(36,0)-(39,9),CHR$(254),5,BF
150 LINE(0,12)-(1,21),CHR$(254),6,BF
160 FOR I=2 TO 7
170    PUT@ ((I-2)*32+2,0)-((I-1)*32+1,79),A%,PSET,I
180 NEXT
190 FOR I=3 TO 7
200    PUT@ ((I-3)*32+194,0)-((I-2)*32+193,79),A%,PSET,I
210 NEXT
220 FOR I=4 TO 7
230    PUT@ ((I-4)*32+354,0)-((I-3)*32+353,79),A%,PSET,I
240 NEXT
250 FOR I=5 TO 7
260    PUT@ ((I-5)*32+482,0)-((I-4)*32+481,79),A%,PSET,I
270 NEXT
280 PUT@(578,0)-(609,79),A%,PSET,6
290 PUT@(610,0)-(639,79),B%,PSET,7
300 PUT@(2,96)-(33,175),A%,PSET,7
310 LINE(0,0)-(639,79),PSET,0,B
320 FOR I=1 TO 19
330    LINE((I-1)*32+33,0)-((I-1)*32+33,79),PSET,0
340 NEXT
350 LINE(0,96)-(33,175),PSET,0,B
360 LOCATE 0,22
```

黒を除く7色のうちの2色の組み合わせですから，$_7C_2=21$とおりの中間色が得られます。

この中間色のなかから，青を背景色にして黄でCHR$(254)を描いた色を，画面いっぱいに埋めるプログラムをつくってみましょう。PRINT文ではなく，LINE文を使います。LINE文を使えば最下行印字後のスクロールを避けることができます。

```
10 WIDTH 80,25:CONSOLE 0,25,0,0:COLOR 0,5:CLS
20 LINE(0,0)-(79,24),CHR$(254),6,BF
30 A$=INPUT$(1)
```

画面全体が中間色になり，何かキーを押すとBASICのコマンド待ちに戻ります。

中間色を作るには，このほかにLINE文を次のように使う方法もあります。

```
10 CLS
20 FOR I=0 TO 638 STEP 2
30    LINE(I,0)-(I,100),PSET,3
40 NEXT
50 FOR I=1 TO 639 STEP 2
60    LINE(I,0)-(I,100),PSET,6
70 NEXT
80 LOCATE 0,10
```

これは，2色の縦の線を交互に引くことにより，中間色を表示するプログラムです。横の線だとドットが大きいので，縦の線の方がよいようです。

§ Returnキーを押すことなく キーボードから1字だけ受け取る方法

(INKEY$, INPUT$(1))

ゲームのプログラムでは，自分の意志をキーボードから伝えるのがほとんどです。高速ゲームでは，キー入力の際にいちいちReturnキーを押さねばならなかったり，キー入力があるまでゲームがストップするようでは実用になりません。そこでReturnキーを押すことなく，キーボードから1文字を受け取る方法を二つ説明しましょう。

一つはINKEY$，もう一つはINPUT$(1)です。INKEY$もINPUT$(1)も関数で

```
A$=INKEY$
```

とか，

```
A$=INPUT$(1)
```

のように使います。A$には押されたキーの内容が文字として入ります。INKEY$とINPUT$(1)の大きな違いは，INPUT$(1)がキーボードから1文字入力があるまで待つのに対して，INKEY$は待ってくれないことです。したがって，高速ゲームではINKEY$を使います。このことは，次の二つのプログラムを実行してみればわかります。

●プログラム1

```
10 BEEP
20 A$=INKEY$
30 IF A$<>"1" THEN 10
```

●プログラム2

```
10 BEEP
20 A$=INPUT$(1)
30 IF A$<>"1" THEN 10
```

プログラム1ではBEEP音が連続して鳴り，1のキーを押したときに止まります。それ以外のキーでは鳴りっぱなしです。

プログラム2では，最初にBEEP音がして後はだんまりです。1以外のキーを押すと再びBEEP音がし，1のキーでストップします。

また，プログラム1のループはファンクションキーの割り込みを受け付けますが，プログラム2では，行番号20でストップしてキー入力を待っているので，ファンクションキーはKEY STOPの状態にあります。

二者択一の質問にYESかNOで答えてほしい場合があります。YESならYキー，NOならNキーを押してもらいたいとき，間違ってそれ以外のキーを押されても大丈夫なようにするには，YとNキー以外を受け付けないようにしなければなりません。それには，次のようにします。

```
10 A$=INPUT$(1)
20 IF A$<>"y" AND A$<>"Y" AND A$<>"n" AN
D A$<>"N" THEN 10
```

押されたキーがY，N以外なら行番号10と20の間をループし，YかNのとき初めて行番号20のあとの文を実行します。

また，1から9までのキー以外を受け付けたくないときは，

```
10 A$=INPUT$(1)
20 IF A$<"1" OR A$>"9" THEN 10
```

とすれば，1から9までのキー以外は先に進みません。

1から9のどのキーが押されたかを実際に1から9までの数字で知りたいときは，

```
A=ASC(A$)-&H30
```

とすれば，Aに値が入り，この後にON GOSUB～などをつけて各処理に飛ばすこともできます。

§乱　数

(RND, RANDOMIZE, INT, TIME)

ゲームに欠かせないのが乱数です。RNDは0より大きく1より小さな乱数を与える関数です。乱数といってもコンピュータはある計算式によって次々に乱数を作り出すわけですから，本当にでたらめな数ではなく，ある規則にしたがった一見でたらめに見える数の列を作り出しています。これはしかたのないことで，本当の乱数を作ろうと思えば，それこそコンピュータ内部でサイコロでもころがすしかないのです。コンピュータの作り出す数の列は擬似乱数と呼ばれ，実用上はこれで充分といえるでしょう。擬似乱数を発生させるにはいろいろな方法がありますが，どんな方法を使っても必ず周期というものがあり，いつかは前に発生した数のところに戻ってきてしまいます。それから後は，前と全く変わらぬ繰り返しということになります。

【例】

1→4→2→8→5→7→1→4→2→8→5→7→

周期　　　　周期

したがって，この周期の長いということが擬似乱数として優秀であることの一つの目安となります。

では，乱数を作ってみましょう。

```
10 FOR I=1 TO 10
20    A=INT(RND*6)+1
30    PRINT USING"##";A;
40 NEXT
```

実行すると，「４２４４１５１６３４」という数の列が得られたと思います。これはサイコロを10回投げるシミュレーションです。プログラムの行番号20を説明しましょう。

RNDを使って発生させた乱数は，０から１の間の数ですからそれを６倍すると，０から６の間の数になります。0.000001や0.999999を６倍すれば0.000006や5.999994になります。サイコロの目は整数ですから，INTを使って引数の値を越えない整数値（小数部分の切り捨て）を作ります。INT(RND*6)は０以上５以下の整数となります。さらに１を加えれば，１から６のサイコロの目ができあがります。

A=INT(RND*6)+1　……１以上６以下の整数

同様に，トランプの１から13を出すには

A=INT(RND*13)+1

とすればよいのです。

さて，このプログラムを何度か実行してみます。結果はいつも「４２４４１５１６３４」となります。どんなでたらめな数の列でも，いつもこのパターンで出てくるとわかっていては，それは一つの規則であり，でたらめとは言えなくなります。

コンピュータの作り出す乱数には周期があると先ほど述べましたが，仮に周期を10としてみましょう。RUNするたびにいつも同じように１番目から出発していると，先のプログラムのようになってしまいます。そこで，たとえ周期があっても出発点を毎回変えてやれば，違ったパターンが得られることになります。あるときは９番目から，あるときは３番目からという具合にします。そんなことができるのかと疑いたくなりますが，内蔵時計の時間を与える関数TIMEを使えば簡単にできるのです。先ほどのプログラムに次の１行を付け加えて実行してみてください。

```
5 RANDOMIZE TIME
```

RUNするたびにTIMEの値が違うので，毎回違った乱数の出発点が得られます。

§もぐらたたきゲームの製作

画面制御やキーボードからの入力について述べてきましたが，最後に応用編として，もぐらたたきゲームを作成してみましょう。

プログラムを作ろうとすると，ほとんどの人が，メモ用紙などを使ってさっそくコーディングに取りかかるか，または，何はともあれキーボードの前にすわってコーディングとプログラムの入力を一緒にやってしまうのではないでしょうか。しかし，これでは非能率です。どんな短いプログラムでも，コーディングしたい気持ちをグッと押さえて，まずは仕様書や流れ図を作るくせをつけておくことが大切です。コーディングに取りかかる前に費やす時間が長いほど，コーディングはスムーズにいき，プログラムも信頼性の高い，すっきりした良いものができます。

プログラムというものは流動的で，作りっぱなしでOKという場合はまずありません。使っているうちに，操作性に対する不満が出てきたり，変化するまわりの環境に合わなくなったりします。追加・変更といったプログラムに手を加える作業が必要となります。そんなとき，プログラムに関する資料が単にLISTだけでは，自分で作ったプログラムでも解析不能といったことになりかねません。そこで，保守性のよいプログラム作りのために，次のことをおすすめします。

【プログラムの作成にあたって】

1．プログラムの仕様をはっきり決めて全体をよく把握する。
　どんなことがやりたいか，どういう入力形式で，どんな形式の出力をするのか，どんな機能を持たせたいのか，をはっきりさせる。さらにプログラムの対象となる事柄（たとえば将棋のプログラムを作りたいなら，将棋そのものに関する知識）自体の完全な認識が必要。

2．流れ図を作る。
　正式なフローチャートでなくともよいから，自分の言葉でわかりやすく書いておく。

【コーディングに際して】

1．構造的に作る。トップダウン方式で作る。
　つまり，全体の流れをまず作り，次にだんだん下位のサブルーチンを作っていく。

2．メインルーチンは短いほどよい。
　この方が全体の流れを把握しやすい。したがって，このように作るためには，メインルーチンはGOSUB～という形を多用する，またそればかりから成るということになる。

3．REM文を多く入れる。
　REM文があれば当然実行速度は落ちる。しかし，ほんのわずかのスピードの犠牲による見返りは極めて大きい。

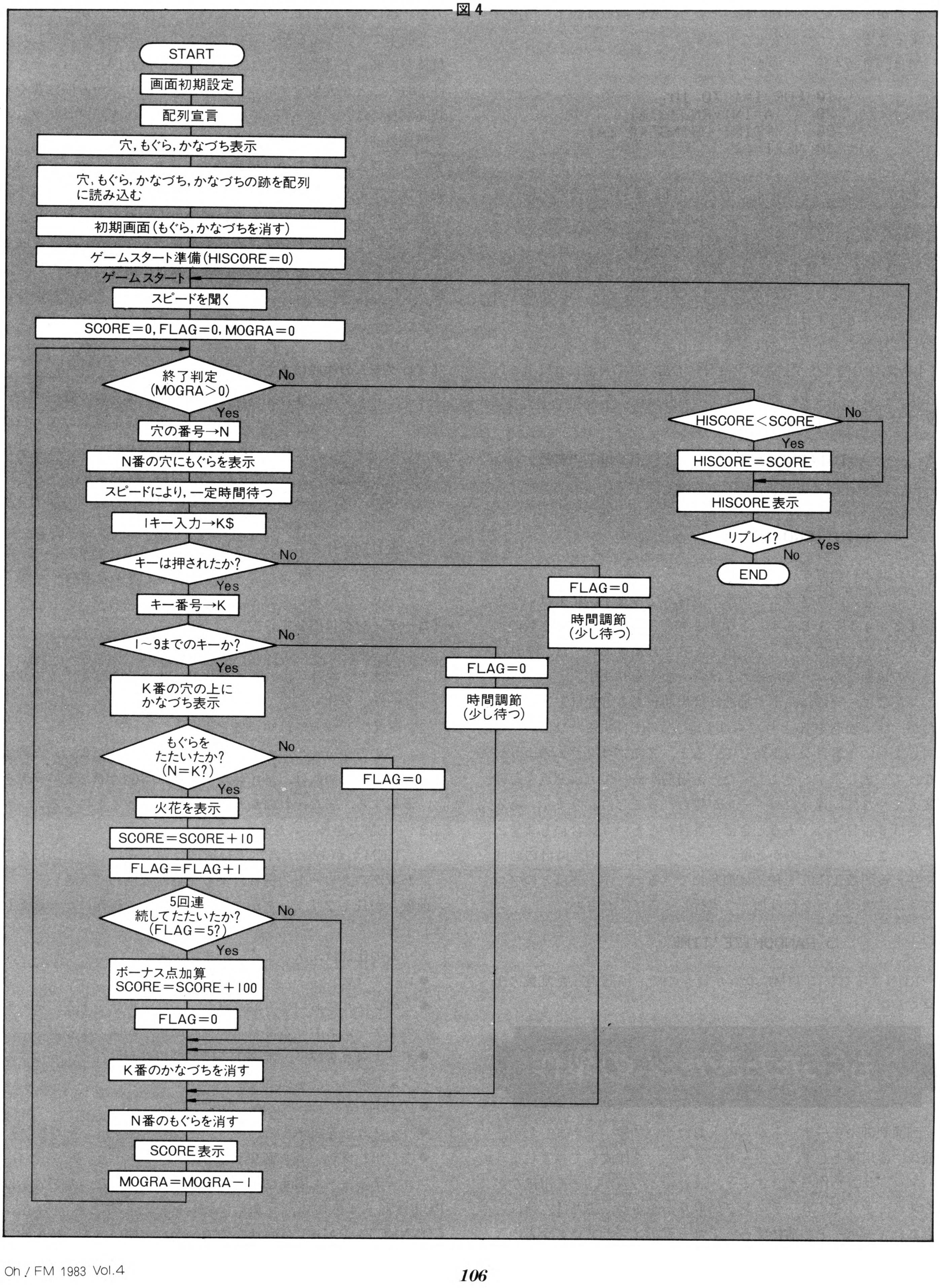
前置きが長くなりましたが，もぐらたたきの製作にかかるとしましょう。

ゲームの仕様は，次のようにします。

- レベル（スピード）を選べるようにする。
- どのレベルにおいても最高点を一つに決め統一する。そのため１ゲームにおけるもぐらの登場回数を40回とする。
- ボーナス点を設け，上達するほど高得点が取れるようにする。つまり高低の差を大きくする。
- FMの特徴を生かすため，表示に凝る。
- ハイスコアを表示する。
- 音を出してゲームを盛り上げる。

フローチャートを**図４**に示します。もぐら，かなづちの表示はPUT@形式４を使い，うまくもぐらをたたいたときの火花はCONNECT，PAINTが高速なので，それを使います。技術的に

図 4


START
画面初期設定
配列宣言
穴，もぐら，かなづち表示
穴，もぐら，かなづち，かなづちの跡を配列に読み込む
初期画面(もぐら，かなづちを消す)
ゲームスタート準備(HISCORE＝0)
ゲームスタート
スピードを聞く
SCORE＝0，FLAG＝0，MOGRA＝0
終了判定(MOGRA＞0)
No
Yes
穴の番号→N
N番の穴にもぐらを表示
スピードにより，一定時間待つ
1キー入力→K$
キーは押されたか?
No
Yes
キー番号→K
1〜9までのキーか?
No
Yes
K番の穴の上にかなづち表示
もぐらをたたいたか?(N＝K?)
No
Yes
火花を表示
SCORE＝SCORE＋10
FLAG＝FLAG＋1
5回連続してたたいたか?(FLAG＝5?)
No
Yes
ボーナス点加算
SCORE＝SCORE＋100
FLAG＝0
K番のかなづちを消す
N番のもぐらを消す
SCORE表示
MOGRA＝MOGRA−1
FLAG＝0
FLAG＝0
時間調節(少し待つ)
FLAG＝0
時間調節(少し待つ)
HISCORE＜SCORE
No
Yes
HISCORE＝SCORE
HISCORE表示
リプレイ?
Yes
No
END

問題なのはボーナス点の判定ですが，5回連続してもぐらをたたいたとき，100点のボーナスをつけることにします。そのための判定用に変数FLAGを用います。すなわち，はじめにFLAGを0にしておき，もぐらをたたくたびにFLAGに1を加えていきます。失敗するとFLAGを0にリセットします。もしFLAGが5になったら5回連続してたたいたことになりますから，ボーナス点100点を加え，再びFLAGを0にします。

実用上まだ問題となるのは，もぐらをたたけなかったときで，このときはいろいろな処理をすべてやらずに次のもぐらを表示することになっています。これではもぐらをたたけなかった場合，すぐに次のもぐらが登場してしまいますから，時間調節（遅延ルーチン）を設ける必要があります。これはFOR～NEXTで空まわりさせればよいでしょう。こうしてコーディングしたのが，**リスト2**です。スピードは一応1～10となっていますが，どんなに大きな数を入れても動作しますので，FM-7ユーザーなどで10でも速いという人は，もっと大きな値を入れて遊んでください。

では，ごゆっくりお楽しみください。満点は1,200点です。

＊　　　＊　　　＊

参考文献

- マイコンライフ，1982年6月号「もぐらたたき」，学研
- FM-8活用研究，工学社
- LEVEL-3　BASIC入門，アスキー出版
- プログラム書法，共立出版
- マイコンピュータ　4号，6号，CQ出版社

リスト2

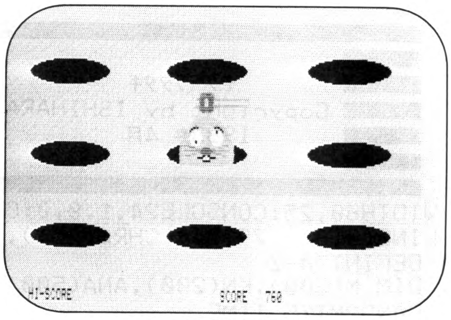
```
10 ' ████████████████████████████████████
20 ' █████                          █████
30 ' █████        ﾓｸﾞﾗﾀﾀｷ           █████
40 ' █████  Copyright by ISHIHARA   █████
50 ' █████       1983年 4月         █████
60 ' █████                          █████
70 ' ████████████████████████████████████
80 WIDTH80,25:CONSOLE24,1,0,0:COLOR0,5:CLS
90 LINE(0,0)-(79,24),CHR$(254),6,BF
100 DEFINT A-Z
110 DIM M(500),KN(200),ANA(500),KNA(220)
120 RANDOMIZE TIME
130 GOSUB 1000:'ｱﾅ,ﾓｸﾞﾗ,ｶﾅﾂﾞﾁ ﾋｮｳｼﾞ
140 GOSUB 2000:'ｱﾅ,ﾓｸﾞﾗ,ｶﾅﾂﾞﾁ,ｶﾅﾂﾞﾁ ﾉ ｱﾄ ｦ ﾊｲﾚﾂ ﾆ ﾖﾐｺﾑ
150 GOSUB 3000:'ｼｮｷｶﾞﾒﾝ(ﾓｸﾞﾗ,ｶﾅﾂﾞﾁ ｦ ｹｽ)
160 HISCORE=0
170 LOCATE5,23:PRINT"HI-SCORE";
180 LOCATE13,23:COLOR6:PRINT STRING$(40,CHR$(254));
190 LOCATE40,23:COLOR0:PRINT"SCORE";
200 GOSUB 4000:'ｽﾋﾟｰﾄﾞ ｦ ｷｸ
210 SCORE=0
220 FLAG=0
230 MOGRA=40
240 'ﾒｲﾝ ﾙｰﾌﾟ
250 WHILE MOGRA>0
260   N=INT(RND*9)+1:'ｱﾅ ﾉ ﾊﾞﾝｺﾞｳ
270   MX=((N+2) MOD 3)*200+100
280   MY=150-((N-1)¥3)*50
290   BEEP1:BEEP1:BEEP1:BEEP0
300   PUT@A(MX-40,MY-23)-(MX+41,MY+8),M,PSET
310   FOR I=0 TO SPEED*100:NEXT
320   K$=INKEY$
330   IF K$="" THEN FLAG=0:FOR I=0 TO 100:NEXT:GOTO 470
340   K=ASC(K$)-&H30
350   IF K<1 OR K>9 THEN FLAG=0:FOR I=0 TO 100:NEXT:GOTO 470
360   KX=((K+2) MOD 3)*200+100
370   KY=150-((K-1)¥3)*50
380   PUT@A(KX-15,KY-37)-(KX+60,KY-24),KN,PSET
390   IF N<>K THEN FLAG=0:GOTO 460
400   CONNECT(KX,KY-25)-(KX+4,KY-28)-(KX+18,KY-29)-(KX+10,KY-31)-(KX+11,KY-36)-(
KX+1,KY-35)-(KX-5,KY-37)-(KX-11,KY-34)-(KX-8,KY-30)-(KX-13,KY-28)-(KX-2,KY-28)-(
KX,KY-25),2
410   PAINT(KX-5,KY-36),2
420   BEEP
430   SCORE=SCORE+10
440   FLAG=FLAG+1
450   IF FLAG=5 THEN SCORE=SCORE+100:FLAG=0
460   PUT@A(KX-15,KY-38)-(KX+60,KY-24),KNA,PSET
470   PUT@A(MX-40,MY-23)-(MX+41,MY+8),ANA,PSET
480   LOCATE47,23:PRINT SCORE;
490   MOGRA=MOGRA-1
```

あそび方

テンキーの1～9までを使います。モグラに対応するキーを押してください。

FM-11の方は，最初にSCREEN　0を実行してください。

```
500 WEND
510 IF HISCORE<SCORE THEN HISCORE=SCORE
520 LOCATE15,23:PRINT HISCORE;
530 LOCATE0,24:PRINT"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ?(Y/N)";
540 A$=INPUT$(1):IF A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N" THEN 540
550 IF A$="y" OR A$="Y" THEN 180
560 WIDTH 80:COLOR 7,0:CLS:END
1000 '■ ｱﾅ ■
1010 FOR I=1 TO 9
1020   READ X,Y
1030   CIRCLE(X,Y),60,7,.15
1040   PAINT(X,Y),0,7
1050 NEXT
1060 DATA 100,50,300,50,500,50,100,100,300,100,500,100,100,150,300,150,500,150
1070 '■ ﾓｸﾞﾗ ■
1080 CIRCLE(100,50),40,6,.6,.45,.05,F:'ﾄﾞｳﾀｲ
1090 PAINT(100,51),6,7
1100 CIRCLE(115,40),9,7,.6,,,F:'ﾒ
1110 CIRCLE(85,40),9,7,.6,,,F
1120 PSET(114,36,0):'ﾒﾀﾞﾏ
1130 PSET(115,36,0)
1140 PSET(116,36,0)
1150 PSET(114,37,0)
1160 PSET(115,37,0)
1170 PSET(116,37,0)
1180 PSET(84,42,0)
1190 PSET(85,42,0)
1200 PSET(86,42,0)
1210 PSET(84,43,0)
1220 PSET(85,43,0)
1230 PSET(86,43,0)
1240 CIRCLE(115,40),12,0,.6,.7,.9:'ﾏﾕ
1250 CIRCLE(85,40),12,0,.6,.6,.8
1260 LINE(67,48)-(133,48),PSET,1:'ﾋｹﾞ
1270 LINE(66,45)-(131,51),PSET,1
1280 LINE(69,51)-(131,45),PSET,1
1290 CIRCLE(90,48),12,0,.4,0,.3:'ｸﾁ
1300 CIRCLE(110,48),12,0,.4,.2,.5
1310 LINE(95,50)-(105,54),PSET,0,B:'ﾊ
1320 LINE(100,50)-(100,54),PSET,0
1330 PAINT(98,52),7,0
1340 PAINT(102,52),7,0
1350 CIRCLE(100,48),9,2,.3,,,F:'ﾊﾅ
1360 '■ ｶﾅﾂﾞﾁ ■
1370 CIRCLE(100,14),10,4,.2
1380 LINE(90,14)-(90,24),PSET,4
1390 LINE(110,14)-(110,24),PSET,4
1400 CIRCLE(100,24),10,4,.2,0,.5
1410 CONNECT(90,17)-(85,17)-(85,21)-(90,21),4
1420 CONNECT(110,17)-(160,17)-(160,21)-(110,21),4
1430 PAINT(100,14),0,4
1440 PAINT(100,24),0,4
1450 PAINT(89,18),3,4
1460 PAINT(111,18),3,4
1470 LINE(98,17)-(102,21),PSET,3,BF
1480 RETURN
2000 '■ ﾓｸﾞﾗ,ｱﾅ,ｶﾅﾂﾞﾁ,ｶﾅﾂﾞﾁﾉｱﾄ ｦ ﾊｲﾚﾂ ﾆ ﾖﾐｺﾑ ■
2010 GET@A(60,27)-(141,58),M,G:'ﾓｸﾞﾗ
2020 GET@A(260,27)-(341,58),ANA,G:'ｱﾅ
2030 GET@A(85,13)-(160,26),KN,G:'ｶﾅﾂﾞﾁ
2040 GET@A(285,12)-(360,26),KNA,G:'ｶﾅﾂﾞﾁ ﾉ ｱﾄ
2050 RETURN
3000 '■ ｼｮｷｶﾞﾒﾝ(ﾓｸﾞﾗ,ｶﾅﾂﾞﾁ ｦ ｹｽ) ■
3010 PUT@A(85,12)-(160,26),KNA,PSET
3020 PUT@A(60,27)-(141,58),ANA,PSET
3030 RETURN
4000 '■ ｽﾋﾟｰﾄﾞ ｦ ｷｸ ■
4010 LOCATE0,24:INPUT"ｽﾋﾟｰﾄﾞ ﾊ ?(1-10) ",SPEED
4020 LINE(0,24)-(79,24),CHR$(254),6
4030 RETURN
```



画面写真

ショートプログラム集

モールス信号プログラム 3題　for FM-7,8,11

Heath K.

アマチュア無線では，モールス信号を使って交信することが多いものです。コンテストでは電話（しゃべって言葉を送る）よりも設備が小さくて済みますし，上級の免許を取るためにも必要とされます。そこで，このモールス信号の練習プログラムなどを3つ紹介します。これまでにも，この手のプログラムは発表されていますが，とにかく短いということで，あえて発表します。これらのプログラムは，FMシリーズのいずれの機種でも実行でき，また他のほとんどの機種でも大丈夫でしょう。

① オートキーヤ

○使用方法

キーを押すと，その文字に対応するモールス信号を出すものです。キャピタルロックで大文字モードにして使ってください。Returnキー，スペースキーは使えますが，間をあける処理（7短点分あけるなど）はしていません。ファンクションキーにメッセージを定義しておけば，メッセージの出力もできますが，そのときにはスペースキーなどへの配慮も必要でしょう。

$\overline{\mathrm{BK}}$，$\overline{\mathrm{AR}}$，$\overline{\mathrm{VA}}$，$\overline{\mathrm{HH}}$は，それぞれ@，・，；，：に対応しています。スピードの変更は行番号130のSPEEDの値を変えて行います（大きくするほど遅くなります）。

和文や$\overline{\text{ホレ}}$は打てません。必要な方は改造してください。簡単な改造で済むでしょうが，データが増えるので僕はやりませんでした。

また，行番号240，260をMOTOR ON，MOTOR OFFとすれば，カセットのリモート端子を無線機のキー入力に接続することで，実際のキーヤとしても使えます。

○入力上の注意など

プログラムの打ち込みにあたっては，特にデータに注意してください。カンマ(，)，ピリオド(．)，アンダスコア(_)の3種類ですが，この文字や数を間違えないでください。アンダスコアは[SHIFT]＋[_ロ]で打ち込みます。同様に230行のピリオドにも注意が必要です。

モールスコードのデータは，ピリオドが短点，アンダスコアが長点となっています。和文を付加される方は，これに合うようにデータを作ってください。

○ここで扱われるプログラムは，個人で利用するほかは著作権法上，無断複製を禁じられています。
COPY RIGHT © 1983 JIPPER

リスト1　オートキーヤ

```
100 '     AUTO KEYER
110 '                    WRITTEN BY JIPPER
120 '--------  INITIALIZE --------
130 SPEED=70 : MAX=ASC("Z") : MIN=ASC("/")
140 DIM M$(MAX-MIN)
150 FOR I=MIN TO MAX : READ M$(I-MIN) : NEXT
160 '------- MAIN ROUTINE -------
170 A$=INPUT$(1) : A=ASC(A$)
180 IF A=13 THEN PRINT ELSE IF A=32 THEN PRINT " ";
190 IF A<MIN OR MAX<A THEN 170
200 C$=M$(A-MIN) : IF C$="" THEN 170
210 PRINT A$;
220 WHILE C$<>""
230   IF LEFT$(C$,1)="." THEN T=1 ELSE T=3
240   BEEP 1
250   FOR I=1 TO SPEED*T : NEXT
260   BEEP 0
270   FOR I=1 TO SPEED : NEXT
280   C$=MID$(C$,2)
290 WEND
300 FOR I=1 TO SPEED*2 : NEXT
310 GOTO 170
320 '--------  MORSE CODE --------
330 DATA _.._.,_____,.____,..___
340 DATA ...__,...._,.....,_....
350 DATA __...,___..,____.,._._.,..._._
360 DATA ........,,,..__..,_..._,._
370 DATA _...,_._.,_..,.,.._.,__.,....
380 DATA ..,.___,_._,._..,__,_.,___
390 DATA .__.,__._,._.,...,_,.._
400 DATA ..._,.__,_.._,_.__,__..
```

② モールス練習〈その1〉

プログラムは，**リスト1**との変更点のみを示しました。**リスト1**を打ち込んだあと，**リスト2**を入力してください。

これは，乱数によってモールス信号を出し，その文字を画面に表示するものです。出力する信号は／，？，$\overline{\mathrm{HH}}$とA～Z，0～9です。信号を聞いて紙に書き，それと画面の表示とを比べて練習します。

このプログラムでは100文字を出力し，一単語を5文字としていますが，行番号130を書き替えることで変えられます。QTYが出力する文字数，WORDが一単語の長さです。

指定した数だけ符号を出力すると，出力の速度（文字／分）を表示し終了します。

リスト2　モールス練習1（リスト1との変更点のみ）

```
130 SPEED=40 : MAX=ASC("Z") : MIN=ASC("/") : QTY=100 : WORD=5
145 RANDOMIZE TIME-INT(TIME/32768!)*32768!
163 ESCAPE$=TIME$ : TIME$="00:00:00"
165 FOR LOOP=1 TO QTY
170 A=RND*(MAX-MIN)+MIN : A$=CHR$(A)
305 IF LOOP MOD WORD THEN 310
306 FOR I=1 TO SPEED*4 : NEXT : PRINT " ";
310 NEXT LOOP
313 PRINT : PRINT QTY/TIME*60;"CHAR/MIN"
314 TIME$=ESCAPE$
350 DATA __...,___..,____.,,
360 DATA ........,,,..__..,,._
```

3 モールス練習〈その2〉

このプログラムは聞こえてきた符号をキーボードに入力して練習を行うものです。50文字の入力を終えると，所要時間を表示して終了します。符号は正解が入力されるまで繰り返し出力され，次の符号へ進みます。出力される符号は，練習1と同じとしました。

リストは，**リスト1**との変更点のみ**リスト3**に示しました。行番号166の値を変えれば，文字数を変更できます。

リスト3　モールス練習2（リスト1との変更点のみ）

```
155 RANDOMIZE TIME-INT(TIME/32768!)*32768!
165 ESCAPE$=TIME$ : TIME$="00:00:00"
166 FOR LOOP=1 TO 50
170 A=RND*(MAX-MIN)+MIN : A$=CHR$(A)
305 Q$=INKEY$ : IF Q$<>A$ THEN 200
306 PRINT Q$;
310 NEXT LOOP
315 COLOR 5 : PRINT : PRINT TIME$ : COLOR 7
316 TIME$=ESCAPE$
350 DATA __...,___..,____.,,
360 DATA ........,,,..__..,,._
```

読者がつくったショートプログラム

リサージュ　for FM-7, 8, 11

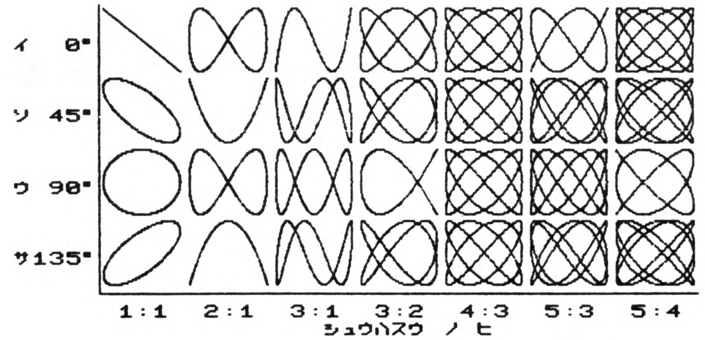
山本元司（福岡市）

これはオシロスコープを扱ったことのある人なら，だれでも知っているリサージュです。念のために説明しますと，リサージュ図形は，オシロスコープの垂直軸に測定したい周波数を，時間軸に基準周波数の電圧を加えると，ブラウン管面に表れてきます。周波数の比，つまりf_1とf_2が1：1で，位相差が0°のときは直線，90°のときは円になります。

行番号160〜220のF1とF2の比をいろいろと変えてみたり，行番号240の位相Tを変えて，模様が移り変わる様子をみてください。また，あまり遅くて困るという方は，行番号260の増分値の分母を小さくするか，LINEを使うか，工夫してください。

プリンタのない方は，行番号380をGOTO 380とでもしてください。

実行結果（HARDC 2）



リスト　リサージュ図形

○このプログラムは，個人で利用するほかは著作権法上無断複製を禁じられています。
COPY RIGHT © M.YAMAMOTO

```
100          ***************  ﾘｻｰｼﾞｭ  ***************
110 PAI=3.14159 : DIM X$(4) : WIDTH 40,20
120 X$(0)="ｲ  0°" : X$(1)="ｿ 45°" : X$(2)="ｳ 90°" : X$(3)="ｻ135°
130 CLS
140 FOR M=1 TO 7
150   ON M GOTO 160,170,180,190,200,210,220
160     F1=1 : F2=1 : GOTO 230
170     F1=2 : F2=1 : GOTO 230
180     F1=3 : F2=1 : GOTO 230
190     F1=3 : F2=2 : GOTO 230
200     F1=4 : F2=3 : GOTO 230
210     F1=5 : F2=3 : GOTO 230
220     F1=5 : F2=4 : GOTO 230
230   FOR N=0 TO 3
240     T=PAI/4*N
250     IF M=1 THEN LOCATE 0,2+4*N : PRINT X$(N)
260     FOR I=0 TO 2*PAI STEP 2*PAI/(400+60*M)
270       X=SIN(F2*I-T)
280       Y=SIN(F1*I   )
290       X=X*36+120+80*(M-1)
300       Y=Y*18+ 20+40* N
310       PSET(X,Y,N+3)
320     NEXT I
330   NEXT N
340 NEXT M
350 LOCATE  6,17 : PRINT"1:1  2:1  3:1  3:2  4:3  5:3  5:4"
360 LOCATE 18,18 : PRINT"ｼｭｳﾊｽｳ ﾉ ﾋ"
370 CONNECT (80,0)-(80,163)-(639,163),2,PSET
380 HARDC 2 : HARDC 1
```

30行ぐらいのショートプログラムを募集したところ，意外に多くの応募がありました。ありがとうございます。しかしはっきり言って，発想の貧困，プログラム技術の未熟など，ちょっとひどすぎるというものも少なくありません。短いだけに，何かこう一つ光ったものを持っているプログラムを希望します。今がチャンス！です。

さて，今回は福岡の山本さんの作品を採用させていただきました。このリサージュ図形，その筋の方にはよく知られているようですが，なかなかきれいなものです。プログラムについて，二，三気のついた点を述べてみましょう。

まず，150～220行は，READ F1，F2で済みそうです。250，350，360の各行は，イニシャライズとして前の方に持っていった方がいいのでは？ それから，山本さん自身おっしゃっているとおり，実行がとても遅いんです。夜寝る前に走らせて，翌朝目が覚めると，いきなり美しい画面が見えてくる，というとてもメルヘンに満ちたテクニックです。個人的には，こういうの大好きです。しかし，実行したらすぐ結果が出るのがコンピュータ，それ以外は認めん，という方もいらっしゃいます。ここは260行のステップ値を大きくして，LINE文を使った方がいいでしょう。ただ，表示する図形が小さく，効果が薄い，ということで，このプログラムになったのでしょう。

なお，FM-11では実行の前にSCREEN 0を，また380行をHARDC 3：HARDC 4としてください。

応　募　要　領

- ■1行80字以内で，できるだけ30行に収めてください。
- ■きれいなプログラムを望みます。マルチステートメントは多用しないでください。
- ■プログラムは，カセットテープまたはディスケットにいれてお送りください。ディスケットの場合は，コピーしたのちお返し致します。
- ■プログラムの説明，使い方など必要なことを簡潔にまとめてください。
- ■プログラムおよび原稿は，多少修正させていただく場合があります。
- ■他誌との二重投稿，盗作など，モラルに反する応募は固くお断わり致します。
- ■応募原稿は原則としてお返し致しません。
- ■応募された方全員に，Oh！FMシールをお送り致します。
- ■すぐれた作品は本誌に掲載し，原稿料にかえてOh！FM特製Tシャツと図書券3,000円分をお送り致します。
- ■原稿にも，住所・氏名・年齢・職業などを明記して，〒102 東京都千代田区四番町2―1 ㈱日本ソフトバンク Oh！FM ショートプログラム集係までお送りください。

READER'S AREA

⇦このFMTシャツを20名にプレゼントします。詳しい応募方法はInformationを見てください。

様なもので申しわけなく思っています。
これからはこのようなことのないようにします。

▶F-BASICユーザーにとって強い味方ですネ。月刊になるともっと魅力的なのに♡ところで，私，FM-7を買ったものの毎日悪戦苦闘しています。バカなことを聞いても笑わない方，文通してください。(○○○市・三門晶子・18)

──このハガキを読んだJ君をはじめとする二，三のスタッフが感動のあまり，晶子さんに電話するという暴挙に走りました。晶子さん，驚かしてホントにごめんなさい。以後はこのようなことのないように，ジステンパーの予防注射を打ち，首輪を付けてつないでおきます。同じようなやからが出ないように，彼女の住所は伏せ字にしました。彼女と文通したいという予防注射済のマジメな方は，編集部あてにお出しください，転送します。だれと文通するかは，晶子さんの判断にまかせます。

▶Oh! FMが週刊になったそうで，おめでとうございます。え，違う，うそでしょう，週刊にしようぜ。編集者一同〝いいとも〟FMの大きな輪を作ろう。(君津市・山口和之・29)

▶第3号になって内容が充実してきたと思う。今のところハードには手を出す気はありません。ペーパーウェアとしての貴誌に期待します。福岡は田舎(マイコンに関して)のせいか，ソフトを全部揃えているショップはありません。(福岡市・田原真一・34)

──しかし九州は「日本のシリコンバレー」といって，業界の注目を浴びているそうですよ。

▶表紙がきれいになりましたね。見つけるのに時間がかかりました。表紙も大切ですが内容も大切です。FM-8の話はもう終りにしませんか？(函館市・井上光円・31)

──3号で，「FM-8の命令を増やそう」という記事がありましたね。アンケートの集計結果，興味のない記事にこれを挙げたのは，ほとんどが7ユーザーの方々であることが判明しました。しかしっ，この記事は7でも有効なのです。具体的な説明はここでは省略しますが，記事のタイトルだけで判断せずに，もっと勉強していただきたいと思うのです。井上さん，8の話は終りに，なんて悲しいことはおっしゃらないでくださいな。

▶1，2号はまったく480円の価値はなかった。でもハード別情報誌という企画をつぶしてはならないとの思いから，涙をのんで買い続けている。3号になってようやくそれがむくいられてきた，という感じをもつに至ったヨ。(新宿区・大矢温・26)

▶マシン語特集が最も役に立った。それからOh! FMにもとうとう戦争ソフトが入りこんできてしまったのは，非常に残念だ。ゲームソフトだけならまだよいが，現実の政治も戦争に向っている。本当の戦争というのはゲームやアニメと違って，本当に人が死ぬのだということをわかっているのか!!(樋川市・福岡秀樹・19)

──福岡さんのお怒り，よくわかります。実は戦争ものを掲載するにあたり，非常に抵抗感があったのです。しかし他に適当なものがなく，一方ではゲームリストを載せてほしい，という声も多く，ご協力いただいたソフトハウスのご好意もあり，ジレンマに悩みながらも掲載してしまいました。まことに中途半端な気持ちから，読者の方に不快感をあたえてしまったことをお詫びいたします。今号も同

▶いや〜，10 FOR I=1 TO X 20 PRINT "第I号" 30 NEXT I のループ1回ごとに確実に良くなってきますね。思わず次の号に期待してしまいます。ぐわんばってえ！(北九州市・福田裕幸・20)

▶第3号の芝野さんの意見は私も相当考えさせられるものでした。確かにハード的に優れたマシンの出現を嘆いてばかりいたのでは，自分の力不足の表明にもなるでしょう。しかしOAが声高に叫ばれ，個人，家庭，企業という様々のレベルでパソコンが普及している今日，その嘆きは別な意味を持ってくると思うのです。ハードに不満な面はユーザーの勉強で補うのはある程度当然としても，それには限界があります。能力，財力，時間など何せホビイストだけがユーザーではないのですから。したがってソフトや周辺機器などハード開発など，8が陳腐化しない環境をユーザーの責任にするのでなく，メーカーやソフトハウスなどが積極的に作っていく姿勢こそ求められるのではないでしょうか。もちろん我々FMユーザーの努力が大前提であることはいうまでもありません。7に不満を言うのは「バカを表明するもの」という思想，一部ホビイストだけでなく，メーカーすらその程度の認識しかもっていないとしたら，FMシリ

ーズに明日はなくなりますよ，ホント。7も8も11もFMユーザーなのですから，もっと建設的，協力的方向をめざしましょう。（江戸川区・須賀隆宏・23）

▶グラフィックソフトパッケージを見て思わず興奮してしまいました。実は僕はロリコンなんです。これからもこういう種類のソフトをどんどん紹介してください。（相楽郡・山口秀康・17）

──最近低年齢のロリコンがやたら流行ってますが，17歳の少年だと相手の女の子は生まれたばかりの赤ん坊になってしまいます。ロリコンというのは37歳のハンバート・ハンバートが12歳のロリータに恋をする，つまり中年おじさんがセーラー服の女子中学生を見て思わずよだれをたらしてしまう，というところに真髄があるのだと思いますが。

▶マシン語のアセンブルリストを載せるのはよいが，だらだらとしたBASICのゲームリストはいただけない。リストを単に載せるより，そのプログラムで使っているテクニックなどの解説を求める。ところで，6809はモトローラ，Oh! FMのマスコットは〝モトモール（motomole）〟。（千葉市・斉藤誠・20）

▶もぐらの名前はタモちゃんがいい。（佐伯郡・高橋英一・51）

▶うーん，中とじが平とじになった。すごい。ページ数がさらに増えることを期待させるような大事件であると言えよう。機械語の特集をしながら桑原岳夫さんの機械語攻撃論も載ったりして。こういう感じというのはとてもおもしろいし，どんどんやるべきことに違いない。（文京区・萩原忠彦・23）

▶各記事，尻切れトンボのようでまとまりがない。（豊橋市・大谷一美・30）

▶FMは使ったことも使う気もないのに，〝3Dパッケージのアルゴリズム〟につられて買ってしまった。ぼくはFMよりPCのほうが好きなのさ。FMなんかどわいっきらいだ（これは単なる偏見です。MZはもっときらいだ）。（練馬区・吉村達志・18）

▶この間，PCを使っている友人に「Oh! FMなんてしょせん隔月刊どまりさ！」とバカにされた。僕は腹がたったので「今年中に月刊になる！」と宣言してしまったのだ！　そこでお願い。今年中に必ず月刊にしてください。（米沢市・村山敏明・18）

▶内容が他の雑誌と比べると理解しやすいが，今すこし深くほりさげて特集してほしい。（横浜市・畠久勝・39）

▶マニュアルだけではわからなかったことが，この本によってわかりやすくなった。今後も多彩な記事を希望します。（岡崎市・本村登・39）

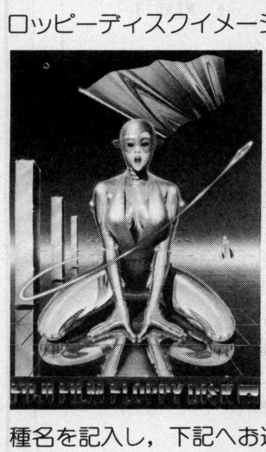
Express

▶Oh! FMの創刊号を1500円分の切手か図書券と取りかえてください。送料そちらもちでお願いします。〒にていつまでも待っています。（〒359 埼玉県所沢市緑町1―12―39―5　安原満）

▶FM-7を8万円でゆずってください。（〒090 北海道北見市小泉577―33 鳳荘　☎01574―24―6519　伊藤弘樹）

Information

▶富士フィルムより，Oh! FM読者30名にフロッピーディスクイメージポスターがプレゼントされます。ご希望の方は，官製ハガキにイメージポスター希望と明示の上，住所・氏名・年齢・性別・電話番号・購読雑誌名（Oh! FM）・使用マイコン機種名を記入し，下記へお送りください。



〒106　東京都港区西麻布2―26―30
富士フィルム（株）磁気材料部
「FDポスタープレゼント」Oh! FM係

締切は7月31日（当日消印有効），当選者には，富士フィルム（株）より直接発送されます。

▶今号44ページでご紹介した商品と，Oh! FMTシャツを読者の方にプレゼントいたします。ご希望の方はとじ込みの愛読者アンケートハガキに必要事項記入の上，希望される商品の番号（商品名の前の番号）を朱記してください。ただし，①〜③はFM-7用です。締切は8月15日（アンケートの締切ではありません念のため）。当選者発表は5号誌上で行います。

①シングルフロッピーディスクセット
FD-7　1名
マイクロシステムズ提供

②漢字ワープロ，FM-JWP/7
SBC提供　5名

③デジ・ソング・エディター　a：クラシック，b：ポピュラー，各5名。a，bを明記してください。
リットーミュージック提供

④PIA基板などのパーツセット（「インタフェースを作ろう」で使用したもの）
亜土電子提供　5名
ただし，拡張バスにディスクなどをつなぐ人は，④aとしてください。

⑤Oh! FMTシャツ　a：赤（M，L），b：ペパーミントグリーン（フリーサイズ）各10枚。a，bの別と，aの場合はサイズもお忘れなく。

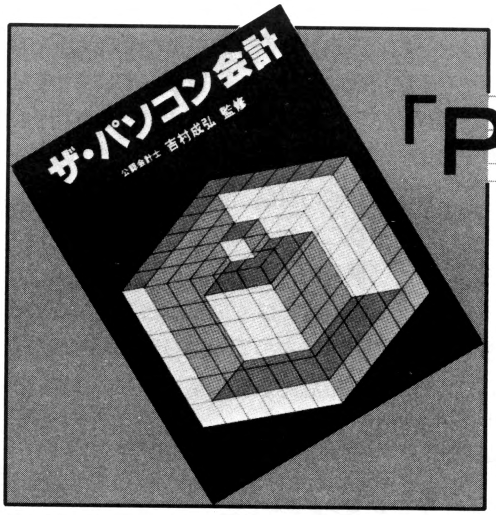
▶創刊号コピーサービスの発送，たいへん遅れて申しわけありませんでした。ただいまエンジン全開で，返送作業をしております。皆さまにご迷惑をおかけしたことを深くお詫びし，作業の迅速化に務めます。ご不明な点はお問い合わせください。なお，こちらに転居先不明などで戻ってしまっている方が何人かいます。心あたりの方，ご連絡ください。

▶このコーナーのために幅広いご意見・ご感想・ご要望などをお寄せください。内容は問いません。イラストなども歓迎です。アンケートハガキに書き足らない場合は，官製ハガキか封書を利用してください。掲載者全員にOh! FMステッカーを進呈。あて先は，

〒102　東京都千代田区四番町2―1
日本ソフトバンクFMエリア係

▶今号より「Express」欄を設けました。売りたい，買いたい，交換したい，仲間がほしい，などの情報をお寄せください。官製ハガキに住所・氏名・電話番号を明記し，FM Exp.係までお申し込みください。取り引きに関しては当事者間で良心的にお願いします。編集部では責任を負いかねます。

▶第5号は9月17日（土）発売予定です。



「PCA財務会計」使用記

公認会計士　田中　高志

コンピュータについては，関連書籍を読むなど一般的な知識はあるものの，自分で操作する機会はなかった。公認会計士という職業上，大手関与先でのコンピュータ利用がどんどん進み，また中小会社からのコンピュータ利用について意見を求められるなど，コンピュータとの係わりが非常に多くなってきた。

このたび，公認会計士吉村成弘先生監修の「ザ・パソコン会計」(第一法規出版発行)を読み，富士通FM-7による「PCA高級財務会計システム」を導入した。

導入してまだ1か月足らずで，小さい会社を2社分処理しただけであるが，事務員(女性)の意見も交えながら，感想やソフトメーカーに対する要望などを述べてみよう。

安くなったコンピュータ

会計事務所の中でも記帳業務を多く行なっている事務所では，5～6年前から，500～1,000万円もするオフコンを導入して，成果を上げているところも少なくない。当事務所は記帳業務が少ないため，ピーシーエー㈱の言うオフコン並みの能力などあまり期待していなかった。なにしろ購入金額が富士通FM-7(CRT, FD×2, プリンタ)とPCA高級財務会計システムの合計で約70万円，3年分割にすると月々約25,000円という安さである。ところが処理内容はもとより，スピードもオフコンとあまり差がないのには正直なところ驚いた。

パソコンは安いが処理が遅いと聞いていた。しかしこの「PCA高級財務会計システム」は，主要部分を機械語で処理しているため，パソコンとは思えない速さである。例えば400～500仕訳の分類に，知人のパソコンは20分程かかると聞いてたが，このシステムでは30秒くらいで終わってしまう。

ともかく安いパソコンが，かつてのオフコン並みになっていることを実感し，またコンピュータの評価はソフトウェアによって大きく変わるものだ，ということを体験したことだけでも大きな収穫である。

PCA財務会計システムの操作性

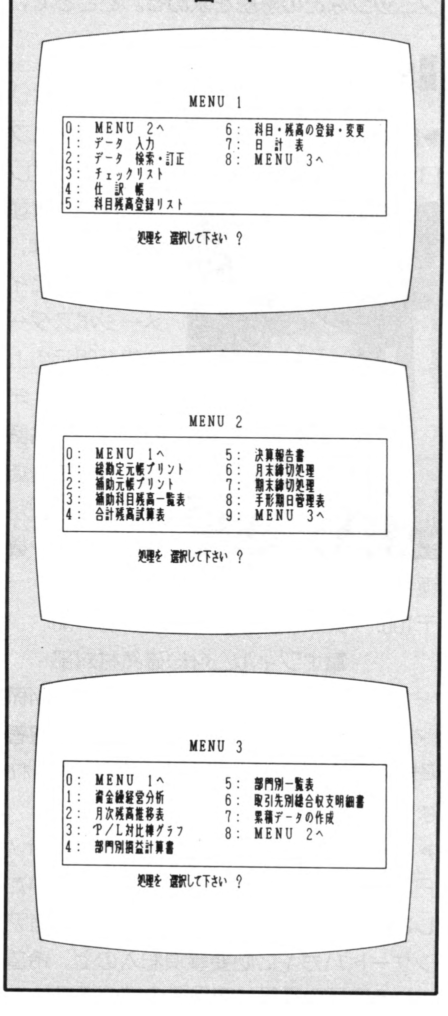
PCAのソフトは，操作仕様書を見ながらだれにでも使え，実際に多くの方々が独学で使っているそうだが，私や事務員は，コンピュータは無論のことタイプライタも使ったことがないため不安である。PCAさんに無理を言って1時間半ばかり説明してもらった。電源スイッチの入れ方から教えてもらったのである。説明を30分程聞くと，これなら使える，という自信がついた。コンピュータを知らなくても，会計が多少わかればだれにでも独学で使えるように実に良くできていた。

具体的に説明すると次のようになる。

(1) プログラムフロッピーをセットして電源スイッチを入れると，自動的に図1のような会計処理の「MENU 1」が出る。「MENU」から処理したい業務の番号を選ぶと，次に何をせよ，という指示が画面に出て，それに従って操作すれば自動的に会計処理ができるようになっている。例えば元帳を作成したいときは，「MENU 2」の「1」を選んで期間指定してやれば，自動的に指定された期間の仕訳が科目別日付順に分類集計されて，プリンタに印字される。

(2) 入出金データが自動仕訳方式になっており，現預金の出納帳からでも入力できる。我々専門家は仕訳を誤ることはないが，これならアルバイトでも処理できる。

図 1



帳表サンプル例

合計残高試算表（損益計算書）—1

自 58 年 1月 1日　至 58 年 1月31日　　単位：円

勘定科目	科目コード	前月残高	借方取引	貸方取引	当月残高
売上高	511	0	8518800	15661543	7142743
ウリアゲ゛ダ゛カ 2	512	0	0	0	0
ウリアゲ゛ダ゛カ 3	513	0	0	0	0
ウリアゲ゛ダ゛カ 4	514～7	0	0	0	0
売上値引戻り高	518	0	0	0	0
純売上高		0	8518800	15661543	7142743
期首棚卸高		0	0	0	0
仕入高	615	0	2016793	0	2016793
シイレダ゛カ 2	616	0	166666	0	166666
シイレダ゛カ 3	617～8	0	0	0	0
仕入値引戻し高	619	0	0	0	0
当期製造原価	625	0	490000	0	490000
期末棚卸高		0	0	0	0
売上原価計		0	2673459	0	2673459
売上総利益		0	11192259	15661543	4469284
役員報酬	711	0	0	0	0
給料手当	712	0	1200000	0	1200000
賞＊＊与	713	0	0	0	0
雑給	721	0	145050	0	145050
退[illegible]	722	0	0	[illegible]	0
[illegible]	[illegible]	0	0	[illegible]	[illegible]
[illegible]費	[illegible]	0	16990	0	[illegible]
販売費及び一般管理費計		0	[illegible]	0	290407[illegible]
営業損益		0	[illegible]	15661543	1565211
受取利息	811	0	0	0	0
受取配当金	812	0	0	12358	12358
ソノタエイガ゛イシュウエキ	813	0	0	0	0
ザ゛ツシュウニュウ入	819	0	0	0	0
営業外収益計		0	0	12358	12358
支払利息割引料	821	0	29307	0	29307
ソノタエイガ゛イヒヨウ 1	822	0	0	0	0
ソノタエイガ゛イヒヨウ 2	823	0	0	0	0
雑損失	829	0	0	0	0
営業外費用計		0	29307	0	29307
経営損益		0	14125639	15673901	1548262

単一振替，複合振替もコクヨの振替伝票のとおり入力すればよい。しかも48科目のワンタッチキーがあるので，科目コードを使わずに日常処理ができる。

(3)　操作キーは「次(PF-3)」「訂正(PF-4)」「終了(PF-5)」と，「Return」の使い方がわかれば，後は画面との対話式で処理ができる。

そのほか摘要コードが登録できるなど会計システムとしての操作性は非常に良いのであるが，難点は摘要欄へカナ文字を入力する際，文字を探すことである。PCAの人が6秒で入力する伝票が1分近くかかる。目下，キー配列に慣れることに懸命である。

汎用性について

会計事務はいろいろな業種に関与している。そのため，先方に提出する資料の書式は銀行や役所にそのまま出せるようなものでなければならない。

その点，PCA高級財務会計システムは応用用紙と専用用紙の使い分けが簡単に行なえる。専用用紙は製造・販売業用になっているため，建築業，学校などには使えないが，科目登録を建築業・学校用にして応用用紙を使うことにより，どんな業種の会計処理でもできるようになっている。しかし，業種別の専用用紙があればさらに良かろう。

また，元帳，試算表，決算報告書などすべて期間指定が日付単位で指示できるので，20日締め切りなども自由に操れ，1枚のフロッピーで2,000仕訳，金額合計1,000億円未満の処理が可能なので，かなり大きい会社でも使える。

科目数は240科目まで登録でき，各科目の残高および仕訳データの発生した科目が自動的に登録科目となり，使わない科目は各種の帳表に出てこないようになっている。

作成できる帳票類

項目	システム A	システム B	システム C
登録データチェックリスト	○	○	○
入力データチェックリスト	○	○	○
仕訳日記帳	○	○	○
日計表		○	○
元　帳（仮締，本締）	○	○	○
補助元帳（仮締，本締）		○	○
合計残高試算表			
B/S（仮締，本締）	○	○	○
P/L（仮締，本締）	○	○	○
資金繰実績表			○
経営分析表			○
部門別損益計算書			○
部門別P/L一覧表			○
月次残高推移表			○
P/L対比棒グラフ			○
手形管理表		○	○
取引先別総合収支明細表			○
決算報告書			
貸借対照表	○	○	○
損益計算書	○	○	○
販売管理費明細書	○	○	○
製造原価報告書	○	○	○
利益処分計算書	○	○	○

補助科目コード

補助科目コードが240種まで自由に使え，小規模の会社であれば取引先別の売掛，買掛金管理を含めた会計処理ができる。

そのほか預り金明細，売上，仕入，手形など補助科目コードを使うことにより，いろいろな管理ができる。だが補助科目名が6文字しか入らないのが少々物足りない。しかしハードとソフト込みで70万円でここまでの処理ができ，さらに5万円足せば部門管理や資金繰り，経営分析ができるというのは，数年前までは信じられないことであった。

まとめ

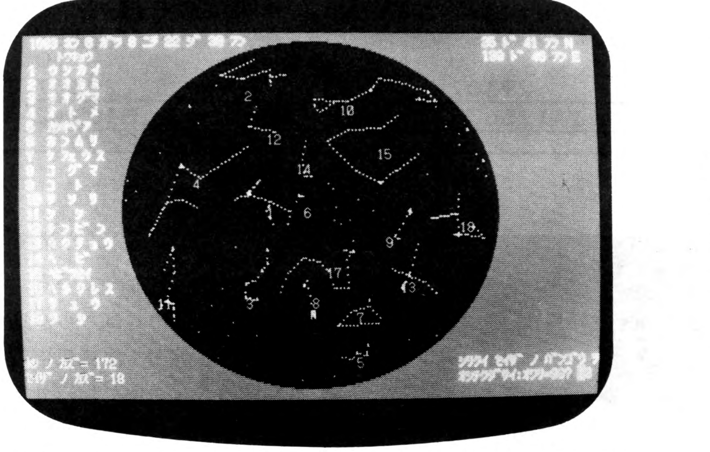
まだよく使いこなしていない段階で述べてみたが，なかなかよくできた会計システムであると感じた。しかし補助科目コードの使い方は操作仕様書だけでは一般によくわからないのではないだろうか。「ザ・パソコン会計」を読まないとわからないのでは少々不親切であると思う。

私はこれを機会に，今後BASICの勉強やパソコンの活用方法を検討しようと思う。皆さんもパソコンをゲームだけでなく，良いソフトを選んで実務に活用されることをお奨めする。

＊　＊　＊

PCA高級財務会計についてのお問い合わせは，【ピーシーエー㈱ 〒150 東京都渋谷区渋谷3-15-5 ☎03(406)3290】まで。

星空への招待 パソコンプラネタリウム シリーズ



■適用機種
FM-7，8／PC-8801，9801

■主要機能
1．世界のあらゆる場所、任意時刻の星空表示
2．日出没時刻、白夜表示
3．星の等級、色に応じた表示、星の名前表示
4．指定星座内の星を点線で連結
5．指定エリアの拡大表示

パソコンプラネタリウムⅠ……機能1～3　¥3,500
パソコンプラネタリウムⅡ……機能1～5　¥5,500
（※フロッピーは1,000円高。）

JIS 日本情報システム（株）
TEL. 03-364-0571
〒160 東京都新宿区高田馬場4-11-8 大都会ビル

役に立つプログラムを作るための

ソフトウェア設計法入門講座 最終回

Computer Science Group　武原　宰

16ビットパソコンが発表されて，ほぼ1年近く経過したが，ビジネスユースに関しては，今や16ビット機以外には考えられない状態である。8ビット機はアミューズメントユースに限られ，低価格化と遊びの要素の追加が進むだろうし，一方，16ビット機は高速化，大容量化の一途をたどり，おのおのまったく別の方向に進化していくことになるだろう。

現在発表されている16ビット機のCPUは，ソフトの蓄積の多さから，インテル系の8086，あるいは8088が圧倒的に多い。しかし，これら80系の16ビットCPUは，16ビットとしては最も初期の頃に開発されたものであるため，後から現れたザイログ社のZ8000や，モトローラ社のMC68000に比較するとかなり能力的に劣っている。中でもMC68000は，16ビットCPUの最高峰に位置するもので，16MHzのクロックで使用した場合の処理能力は，現在の32ビットスーパーミニコンを凌ぐと言われている。このCPUを採用したパソコンは，従来のパソコンの概念をはるかにこえて，汎用機並みの能力をもつことになる。

実は，この68000を採用したパソコンがそろそろ市場に現れだした。80系の16ビットがようやく出揃ったばかりというのに，早くも次の段階へ移ろうというのだから，ユーザーとしてはたまらない。この傾向は今後も続いて，2年後くらいには32ビットパソコンとなり，もうこれ以上処理能力が高まってもパーソナルユースとしてはやることがない，というところまで行きつかないと止まらないだろう。

ところで，ほとんど無限に拡大していく処理能力を，われわれユーザーは何に役立てればいいのだろうか。

一方でパソコンにやらせたい仕事は山ほどあり，他方ではパソコンの処理能力はどんどん高まる。今まではパソコンにやらせるのは無理，と思われていた仕事が無理でなくなりつつある。この間をスムーズにつなげて，パソコンにやらせる仕事が増せば万々歳であるが，実際にはそう簡単にはいかない。本来，何にでも使えるというコンピュータに，何か特定の仕事をさせようと思うと，その仕事とコンピュータを結びつける周囲の準備にたいへんな手間がかかることとなる。それは周辺機器の選択から始まり，プログラムの設計製造はもちろんのこと，そのもっと根源にまでさかのぼって，コンピュータにやらせようとする仕事そのものの分析にまで進まなければならなくなる。これらの作業を総合してシステム設計と称するわけだが，このシステム設計こそコンピュータを活用する上で，単なるプログラム設計や製造よりはるかに大切で，また難しい仕事なのである。今回は，システム設計について少し考えてみることにしよう。

I. システム設計の重要性

従来のパソコンの世界では，マシン語のダンプリストをそのまま読み下せるような達人がプロ中のプロと思われていたが，ビジネスユースの世界で最高のエンジニアは，SE（システムエンジニア：システム設計を担当するエンジニア）である。最近はやたらにSEという言葉が氾濫しているが，日本にはSEという名に値するエンジニアはまだいない，と言い切る人もいる。一人前のSEとなるにはコンピュータやプログラムに対する知識はもちろんのこと，それ以上にコンピュータにやらせようとする業務そのものに精通し，しかもその業務をコンピュータ化できるような整備技術を身につけている必要がある。それは，最終的に役に立つシステムを作り上げるには，その対象となる業務を熟知し，分析しつくした上でコンピュータ化しなければならないからだ。この段階の仕事，すなわちシステム設計の成否がシステム全体の成否にかかわっており，システム設計に誤りがあると，それ以後のプログラム設計などの工程でどんなに優れたテクニックを駆使しても，取り返しがつかなくなる。

また，SEの仕事は単にコンピュータに対するのみでなく，コンピュータを導入することによって最大の省力効果を発揮するように，人間組織を組み直すことにも向けられる。従来の仕事のやり方の再検討，オペレータの教育方法の検討なども，SEの重要な仕事である。

このようにSEには，ハードウェアとソフトウェア（この場合は人間の扱いをも含めた広い意味でのソフトウェア）両面にわたっての幅広い知識と経験が必要とされる。ちなみにIBM社の公認SEとなるには，睡

眠時間3時間で，あらゆる分野にわたり3年間猛勉強を続け，数十度におよぶ試験をすべてクリアしなければならない，と言われている。SEになる努力もたいへんだが，SEを養成するためにIBM社が費やす費用も莫大なものである。

II. 業務分析のコツ

さて，SE作業の中でも最も大切な，業務分析について考えてみよう。職種は多種多様であっても，基本的な注意点は同じだ。

1 目的を明確にする

コンピュータに何をさせるかという以前に，何のためにコンピュータを用いるのかを考えてみる必要がある。

最近，ブームにのってパソコンを買ってはみたものの，結局やらせる仕事がなくてほこりをかぶっているという話をよく聞く。中古ショップに売り物が絶えない，という現象がこれを裏づけている。コンピュータという西洋合理主義の極致のような機械は，使用目的をはっきりさせない限り，決して充分な働きはしてくれない。

コンピュータを使用する目的は，大きく分けて二つある。一つは，今，人間がやっている仕事をコンピュータにやらせて，いかに人間が楽をするかというい わゆる省力化であり，もう一つはそれとは逆に，人間だけではとてもできないような高度な仕事をコンピュータの導入により実現しよう，という場合である。前者の場合にはいかに省力化の効果をあげるかということが目的であるから，たとえばコンピュータの導入によってデータ入力の作業が増え，その作業が導入前よりも多いとすると，何のための導入なのかわからなくなってしまう。逆に後者の場合であれば，多少入力の手間が増したとしても今までとてもできなかったことが可能になるのであれば，導入の効果はあった，ということになる。このあたりの見きわめが大切である。

2 業務をIPO化せよ

本講座を通読いただいている方にはおわかりだと思うが，IPOとはInput-Process-Output，入力－処理－出力の意味である。

業務のIPO化とは，業務を入力－処理－出力に分けて整理することである。これは繰り返し説明したように，コンピュータを用いる上での鉄則であり，大はシステム設計から，小はサブルーチン作成にいたるまであらゆるレベルに通用する。なぜならコンピュータとは本来，入力データを取り込み，それを処理し，結果を出力するというだけの単純な機械だからである。単純な機械に効率よく働いてもらうためには，この機械に最も適した形で仕事を与えてやることが早道である。

したがって，まず目的が決まると次に，出力を決め，何を結果として得たいかをはっきりさせる。出力が決まると，その出力を得るために必要な処理と入力が必然的に決まってくる。ここでは，出力→処理→入力の順に決めていくのがコツだ。その逆ではないことに注意されたい。

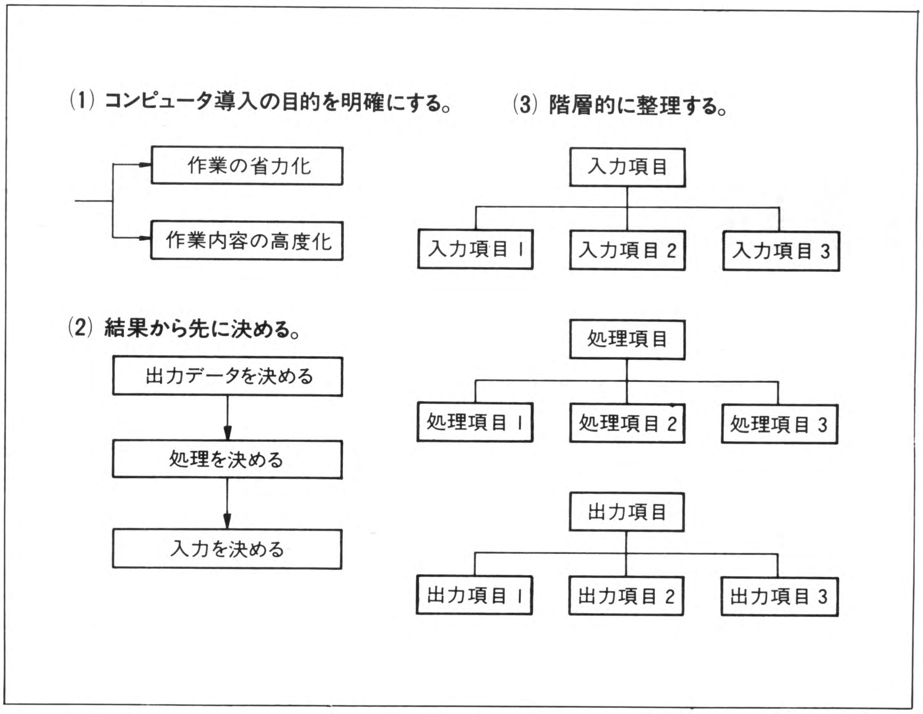
失敗したシステム開発は，往々にしてこの逆の過程をとっている。すなわち，まず何らかの入力データがあり，これをコンピュータに処理させたい。どんな効果があるかはっきりわからないが，コンピュータにほうりこみさえすれば何かすばらしい結果が出てくるに違いない，という曖昧な目的でスタートする。入力データに応じて何か適当な処理を考え，その結果として出力データが出てくる。当然その出力は，目的や効果を充分検討して決められたものではないために，役に立つ場合はまれであるということになる。

結果を先に考えないと，コンピュータは使いこなせないのである。

3 項目を階層的に処理せよ

入力－処理－出力の各レベルにわたってコンピュータが扱うべき項目は多い。それらを効率よく処理し，しかもその流れを人間から見てわかりやすくするためには，多岐にわたる項目を誰にでもわかる形に整理することが必要である。わかりやすく整理するには，グループに分けるのが最も確かな方法で，似た項目同士をグループにし，そのグループ間を結ぶ関係をはっきりさせる。できればグループをさらにレベル（階層）に分けて，ハイアラキ（階層構造）にすることが望ましい。

この階層構造化ということも前号までで解説したソフトウェア設計法，HIPO法に共通した考え方であり，コンピュータに関係した何かをやろうとする場合には，不可欠な考え方である。



III. 変更に耐えるシステム

システム設計段階で項目を階層的に整理するというのは，きわめて重要な意味をもっている。階層構造の良否がシステムの柔軟性を決定するからである。

1 システムの柔軟性

ここで，システムの柔軟性ということについて説明しよう。

コンピュータシステムには，仕様の変更がつきものである。小は，入出力項目の追加や計算方法の変更から始まり，大は出力機器の追加やディスクの増設，さらには，CPUの置き換えということまでありうる。システム設計の方法が悪いとこれらの変更の一つ一つに対して，ソフトウェア変更のために大変なコストがかかることになる。柔軟なシステムとは，これらの仕様変更を最少のコストで吸収しうるシステムのことである。

2 ソフトウェア危機

いったん作り上げたシステムを少しでも改造しようとすると，たいへんな費用がかかるという問題は，大型機の世界ではすでに，1960年代末ごろから大きな問題となっていた。ちょっとした変更でもその変更で影響を受けるプログラムの数が多いと，たいへんな作業量となり，場合によっては初めから作り直した方が安くすむ，というケースが頻発した。そしてシステムを初めに作り上げるときよりも，その後，運用していくのに要するコストの方がはるかに高くついてしまう，という事態があちこちで起きたのである。これは，ソフトウェア危機(software crisis)と呼ばれ，特に世界で最も大規模なシステム開発を行っていたアメリカ国防省で問題となった。実は先号まで解説してきたHIPO法は，この危機を切り抜けるためにIBM社が開発した，総合的なソフトウェア設計法なのである。

3 柔軟性を実現するコツ

残念ながら現段階においては，決定的な方法は確立されていない。それどころか，ハードウェアの進歩が，ますます巨大なシステムとソフトウェアを可能としたために，危機はますます深まりつつあると言える。銀行オンラインなどの巨大なシステムでは，中途での変更を完全にあきらめ，完成後何年間かは，極力仕様変更を避けて運用し，機を見て一挙に新しいシステムを作り直すという方法をとっている。早い話が使い捨てである。

パソコンの世界でも事情は同じようなものである。

ある部署にきわめて優秀な人がいて，すばらしいシステムを作り上げてもその人が配置転換となりその後，組織が変わってシステムへの入力項目に変更があったりすると，もう誰も手をつけることができなくなり，結局捨てられることになる。

さて，絶望的なことばかり並べても仕方がないので，それなりの方法も少し考えてみよう。

●できる限り小さな項目に分割すること

小さな項目をいろいろに寄せ集めれば，様々な形を得ることができる。大きいとそうはいかない。初めから大きな項目を単位として扱っていると，あとで融通がきかなくなる。

●項目間の関係をできる限りなくす

項目間に特別な関係があると，自由に組み合わせることができなくなる。これは，要素間の独立性と呼ばれる概念で，システム設計を行っていく上で大切なキーポイントである。各項目間の関係を見抜くには，その対象となる業務に精通していることが必要であり，SEの真価が問われる場面である。

IV. マニュアルを先に書く

前述したように，結果から先に考えるのがシステム設計のコツである。これをつきつめると，通常は最後に書かれる操作マニュアルを，いちばん先に書いてみるという方法が浮かんでくる。

事実，システムの良否を決定するのは操作性であり，マニュアルを先に書くことによってそれが見えてくる。特に多くのシステムにおいてネックとなるのは，入力の部分である。

いかにコンピュータの処理が早くても，人間による入力に多大の時間を要したのでは，コンピュータを導入した意味がなくなる。

入力の手順を細かく記述したマニュアルを先に書いてみると，このあたりの事情が明らかになるのである。

最近のアメリカなどにおけるシステム開発では，先に完全なマニュアルを作ってから，ソフトウェアの製造にかかるケースが多くなっている。

V. ハードウェアの選択

冒頭にも述べたように，ハードウェアの進歩は，あいかわらずすさまじい勢いで進んでいるが，いずれはその機能と価格によって，はっきりとした階層に分れるようになるだろう。アミューズメントを中心とした8ビット機は10万円以下に，スモールビジネス用の16ビット機は30万〜40万円台に，そして本格的なビジネスユース，および科学技術計算向けの32ビット機は300万円前後の価格に落ち着くものと考えられる。

現状ではビジネス用にパソコンを選ぶなら，新しいものほどよい，ということになるが，上述のようにその階層化が進むと，自分のビジネスに適した階層を選択するというのが重要なポイントになってくる。

さらに周辺装置も，カラープリンタや超高解像度，多色表示のディスプレイなどが

低価格で登場して，選択の幅はますます広がることだろう。

いずれにせよ，まず，コンピュータの導入によりどのような効果を求めるか，ということを先に考えて選択することが大切だ。

VI. ソフトウェアの選択

ミニコン以上の世界では，コンピュータを使う上で最もお金のかかるのは，ソフトウェアの部分と言われている。ソフトウェアは人海戦術で作る以外に方法がないため，たちまち膨大なコストとなってはね返ってくる。

ミニコンシステムの場合でも，初めから作ると1システムあたりの開発に数千万円かかるというのが相場である。ところが現在では，2～3年前のミニコンをはるかに凌ぐパソコンが100万円以下で手にはいる。100万円のハードウェアに数千万円のソフトを作るなどとは誰も考えないだろう。

さらには前述したように，せっかく膨大な投資をして作り上げたソフトウェアも仕様変更を前提にし，使い捨てになるケースが非常に多い。どうせ使い捨てるのなら初めから開発などせずに，既製品のパッケージソフトを購入して仕様が満たなくなれば，また別のパッケージを買えばいい，ということになる。実際，初めから開発した場合とパッケージを購入した場合のコスト比は，数百倍近くはあると思われる。数百回使い捨ててもペイするわけである。

ところがパッケージはたいてい帯に短したすきに長しで，自分の仕事にぴったりのものはまず存在しない。したがって，自分の仕事をパッケージに合わせるか，さもなければパッケージを改造して使うということになる。少しソフトのわかる担当者のいる会社では，パッケージを一部改造して使用しているが，たいていのパッケージはその内部構造を公開していないため，改造にはかなりの労力を要する。今後はパッケージとオーダーメイドの中間に位置する，改造可能なパッケージとでもいうものが販売されると，ユーザーとしては助かるだろう。

もっともソフトウェアメーカー側としては，設計書の公開などにはいろいろ問題があるだろうが，ユーザーの立場に立てば，容易に改造可能なパッケージの登場を望みたいところである。

4回にわたって連載してきたソフトウェア設計法入門講座は，今回で一応終らせていただきます。

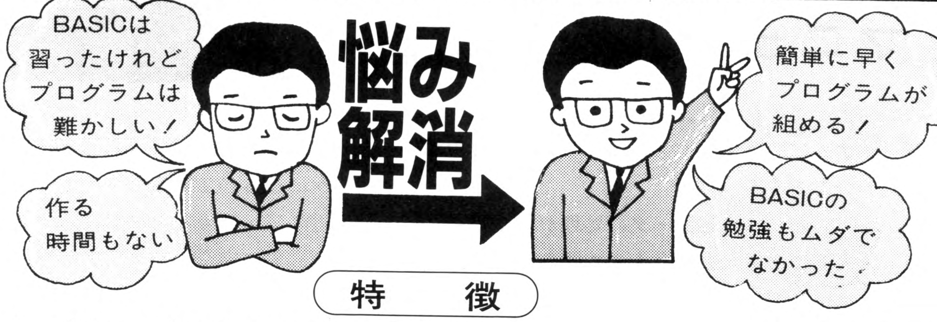
ご愛読，ありがとうございました。

好評新発売

あなたも今日からプログラマー!!

BEST ベスト
FOR BUSINESS EVERY PURPOSE SOFTWARE TOOL

ビジネスユースの為の簡易ソフトウェアツール!!



適応機種
- 富士通FM-7
- 富士通FM-8
- 富士通FM-11
- NEC PC-8800
- NEC PC-9800

特徴

1. 難しいプログラムをパーツ化!
   - 難しいプログラムの部分をパーツ化してフロッピーに格納してありますからお客様は自由に取り出して、つなぎあわせ、必要な変数やデータを書き込むだけで、本格的なビジネスプログラムが作成できます。
   - パッケージソフトや、汎用ソフトと違い、内容は全くのハンドメイドですから利用者の思想が充分盛り込め、拡張性があります。

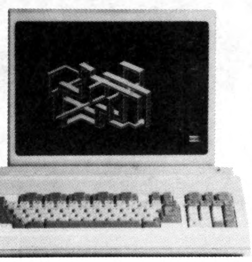
提供媒体 ●フロッピィディスク（5インチ・8インチ）

※上記に関するお問い合わせは企画部 福岡・柘植まで。

サンコンピュータスクール・サンコンピュータ総合研究所
理工産業株式会社
太陽グループ
〒510 三重県四日市市九の城町4-20
TEL〈0593〉51-1651(代)

ウチダマイコンスクールチェーン教室
- 四日市教室 〒510 三重県四日市市九の城町4-20 ☎(0593)51-1651
- 津教室 〒514 三重県津市本町23-17 ☎(0592)25-6251
- 松阪教室 〒515 三重県松阪市船江町785 ☎(0598)51-4312

グリーンモニター　¥25,000～・カラーディスプレイ特価



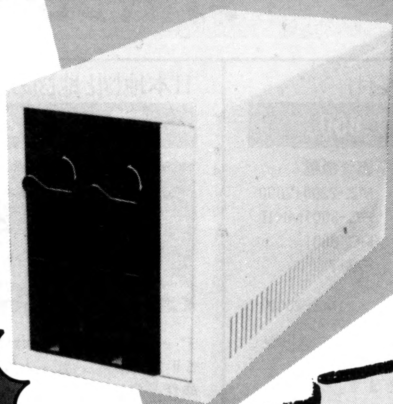
¥126,000

¥398,000

FMデータレコーダー	MB27501	¥12,800
FM－7　漢字ROMカード	MB22405	¥35,000
FM－7　フロッピーインターフェースカード	MB22407	¥14,800
FM－7　RS232Cインターフェースカード	MB22406	¥15,000
FM－7　Z80カード CP/M-80	MB28021	¥35,000
FM－7　日本語ワードプロセッサー		¥39,800
FM－7　FBASIC		¥ 7,000
FM－8　ディスクユーティリティー		¥11,000
FM－8　倍精度関数ライブラリー		¥15,000
FM－8　日本語入出力ライブラリーVOL 1		¥ 9,800
FM－CALC		¥68,000
FM－FBASICコンパイラー		¥33,000
COMSOL Compiler 5″DISK版		¥29,800
FM-11 漢字ROMカード	MB28111	¥46,000
FM-11用α-CARD（68008カード）		7月中旬発売予定

FM-7・8用フロッピードライブ

THIN LINEデュアルドライブ



○STD（自作派の方に）
¥99,800

○FM-7set
（フロッピーアダプター
マスターディスケット・ケーブル付）
¥141,000

○FM-8set
（フロッピーアダプター
マスターディスケット・ケーブル付）
¥143,000

店内 プリンター 大量デモ中！

ボーナスセール プリンターフェア

7月末日まで

期間中5万円以上お買上げの方に
オリジナルTシャツプレゼント！

ADO マイコンブックショップ

特報

5″ディスク版

○ミッションアステロイド
¥7,800

○ミステリーハウス
¥8,800

○ウイザード＆プリンセス
¥12,300

その他各種テープ版多数

03-255-9515

千代田区外神田2-2-1

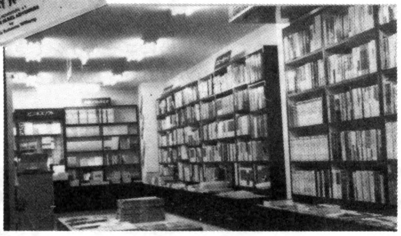
1F・マイコンショップ
2F・コンピュータ書籍
・半導体データブック他

その他 FM関係各種有！

・ディスケット
・カセットテープ

お得です

特　価　中

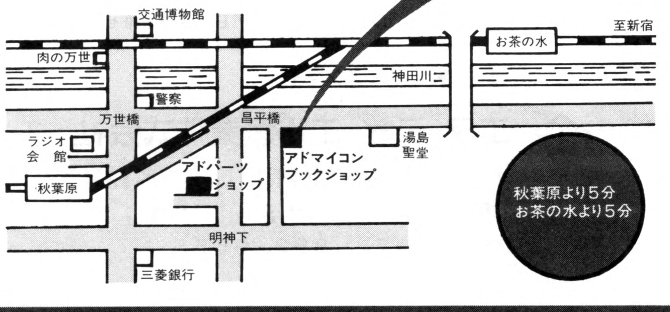


2F

・FM関係書籍
・各種コンピューター書籍
・半導体データブック
・電気・電子工学
・オーディオ関係

3000種以上

ディスケットの通信販売も取扱中です。



通信販売も店頭特価にてクレジット／ローンでお求め頂けます。

お問い合せは **03-255-9515** へお気軽に

店頭にてローンご希望の方は身分証明（免許証等）と
印鑑をお持ち頂ければ10分程で、お持ち帰りになれます。

クレジットご希望の方は、1～36回まで、均等払い又はボーナス併用もできます。

1回	3回	6回	10回	12回	16回	18回	20回	24回	30回	36回
3%	5%	7%	10%	11%	15%	17%	18%	21%	25%	30%

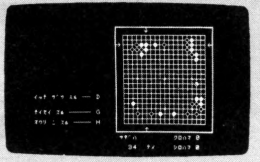
学校・官公庁納入実績豊富!!所定の様式にて受承ります。担当：今徳・雲足257-1029へ

ユニークなオリジナルソフトで躍進する

CHAMPION SOFT

好評発売中ソフト

囲碁トレーナー(マニュアル付)
囲碁道場／各￥3,800



適合機種
MZ-2200・2000
PC-8001MKII
PC-8801
FM-7

(MZ-2000)

棋譜や定石をコンピュータに入力し、一手ずつ再現して見たり、入力した棋譜や定石をテープにSAVEしておき、後日再現して見ることもできます。
●原作／S.Taniguchi

日本地図と地図あてゲーム(マニュアル付)
マップ-47／各￥3,800

適合機種
MZ-2000・80B
PC-8801
PC-8001MKII
FM-7

(PC-8801)

データ数3100個に及ぶ精彩グラフィック日本地図による県別地図の勉強と大人も楽しめる地図あてゲーム。
●作／白木善喜

優勝チーム予想ゲーム
熱戦甲子園／各￥3,800

適合機種
PC-8001MKII・8001
FM-7
MZ-2000・80B
パソコンテレビX-1

(FM-7)

チームが集まり草野球。持点を賭けて最後のチームを予想。ゲーム＋集計プログラム
●作／ビックスマイコ

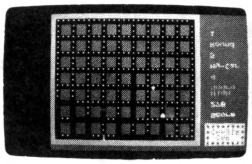
人生シミュレーションゲーム
ザ・サクセスマン／各￥3,500

適合機種
MZ-2000
FM-7
PC8001MKII
パソコンテレビX-1

(FM-7)

君の人生は平穏無事か。それとも波らん万丈か。
君の人生をシミュレートしてみよう。
●作／白木善喜

反射神経型BASIC＋マシン語高速ゲーム
カーチェイサー／各￥3,500

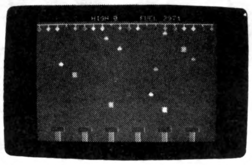


適合機種
MZ-2000・80B
MZ-700
PC-8001MKII・8001
パソコンテレビX-1
FM-7、JR-200

(MZ-700)

コンピュータがコントロールする車の追跡をうまかわしながら、路上のドットを喰い逃げできるか……。
●原作／谷田宣文●移植／籔内雄美

反射神経型BASIC＋マシン語高速ゲーム
ジャンピングクロス／各￥3,500



適合機種
MZ-2000・80B
MZ-700
PC-8001MKII・8001
パソコンテレビX-1
FM-7、JR-200

(MZ-700)

Jumperを操り、右端から流れてくるMeteor(あるいはUFO)を避けながら蔓に下っているダイヤモンドをいくつとれるか……。
●原作／谷田宣文●移植／籔内雄美

シミュレーションビジネスウォーゲーム
コンピュータウォーズ／各￥3,500

適合機種
PC-8001MKII・8001
MZ-2000・80B
FM-7、MZ-700
パソコンテレビX-1

(PC-8001MKII・PC-8001)

君はコンピュータ会社の営業部長。現在10%のシエアを3年後に50%にできるか……。
●作／白木善喜

シミュレーションウォーゲーム
コンバット-47／各￥3,500

適合機種
PC-8001MKII・8001
MZ-2000・80B
FM-7、MZ-700
パソコンテレビX-1

(FM-7)

1～4人で楽しめる知的戦略ゲーム
47か国のうち君は何か国を占領できるか。
●作／白木善喜

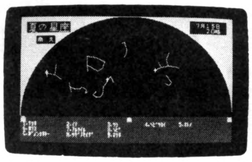
コンピュータダイスプレイゲーム
ウエディングレース／各￥3,500

適合機種
PC-8001MKII・8001
MZ-2000・80B
FM-7、MZ-700
パソコンテレビX-1

(パソコンテレビX-1)

君はめでたく結婚にゴールインできるか。それとも破産のうきめにあい、地獄行となるか……。
●作／白木善喜

星空の旅
ザ・プラネタリウム／各￥3,500



適合機種
PC-8801
PC-8001MKII
MZ-2000、FM-7
パソコンテレビX-1

(パソコンテレビX-1)

北天・南天別に四季の星座を満喫できます。
●作／白木善喜

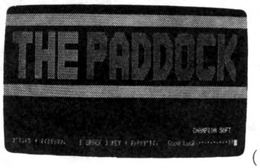
楽しい星座教室(マニュアル付)
ザ・スターズ／各￥3,800

適合機種
PC-8801
PC-8001MKII
MZ-2000・80B
FM-7
パソコンテレビX-1

(FM-7)

北天・南天別に月，日，時刻を変えてインプットすると各星座の動きを見れます。
また、星座を拡大して見ることもできます。
●原作／館盛光弘●移植／北村隆次

科学的競馬予想学(マニュアル付)
ザ・パドック／各￥3,800



適合機種
PC-8001MKII・8001
FM-7・8
MZ-2000・80B
パソコンテレビX-1
JR-200

(パソコンテレビX-1)

①持ちタイム②前走タイム③脚質～等14種のデータを入力して、勝馬を予想し連複券を推せんします。
●作／ビックスマイコ

制作 株式会社チャンピオンソフト
〒530 大阪市北区西天満6丁目1番12号瑞穂ビル
TEL(06)365－9900
ファックス(06)361－1176

発売元 エヌ・テイ・エル産業株式会社
〒657 神戸市灘区篠原北町1丁目5番13号
TEL(078)881－1914
ファックス(078)882－0555

ユニークなオリジナルソフトで躍進する

CHAMPION SOFT

自信の新作ソフト続々登場

ゲームと絵本がドッキング

むかしむかしゲームシリーズ 新発売

各¥3,800

むかしむかしゲーム① 桃太郎

適合機種
PC-8001MKII
PC-8801
FM-7
MZ-2200

(FM-7)

イヌ、サル、キジをおともにして鬼退治。無事宝物を持って帰れるか。

●原作／小嶋敏之
●移植／安井高次

むかしむかしゲーム② 浦島太郎

適合機種
PC-8001MKII
PC-8801
FM-7
MZ-2200

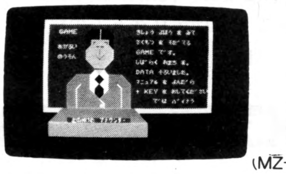
(FM-7)

あなたは浦島太郎を背中にのせた亀になり、無事竜宮城にたどりつき、タイやヒラメの踊りを見ることができるか。

●原作／小嶋敏之
●移植／安井高次

インテリジェントシミュレーションゲーム

食糧パニック／各¥3,800



適合機種
MZ-700
PC-8001MKII
FM-7
パソコンテレビX-1
PC-8801

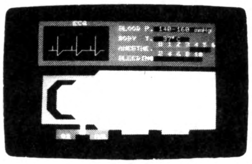
(MZ-700)

西暦200X年。天候異変により大凶作が数年続き食糧パニックが発生。知恵をしぼって食糧増産に努力されたい。

●原作／平岩徳一

インテリジェントシミュレーションゲーム

Dr.麻酔科医／各¥3,800



適合機種
MZ-700
PC-8001MKII
FM-7
パソコンテレビX-1
PC-8801

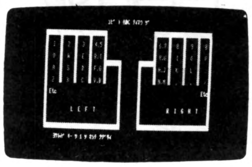
(MZ-700)

血圧、体温、麻酔レベル、出血量、心電図を見ながら、無事手術を完了させることができるか。

●原作／平岩徳一

タイプトレーナー

タイプ練習／各¥3,800



適合機種
PC-8001MKII・8001
MZ-2200・2000,80B
FM-7
パソコンテレビX-1
MZ-700、JR-200

(PC-8001MKII・8001)

基礎コースではアルファベットの練習、実力養成コースではBASIC用語等を練習できるタイプトレーナー。

●原作／森泉平助

NewGame 君はゲームの中のゲームを見たか？

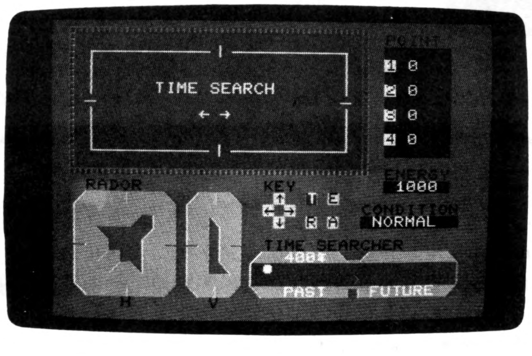
TIME PATROL／タイムパトロール

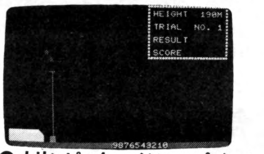
リアルタイム・インテリジェントゲーム

ナ・ナント！ゲームの中にゲームが4つもあるのだ。宇宙のパトロール隊員になるためには、4つのゲームすべてにチャレンジしなくてはならない。

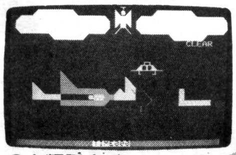
●写真はMZ-700
各¥3,800

新発売

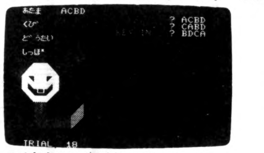




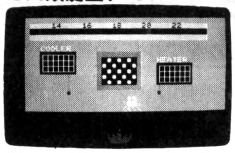
●オリンピックハイジャンプゲーム



●人類誕生ゲーム



●遺伝子組替えゲーム



●お天気ゲーム

■適合機種：MZ-700、PC-8001MKII・8001、FM-7、パソコンテレビX-1、MZ-2200■原作　平岩徳一■移植　夏目明彦，CAT'SEYE

⊙お求めは、お近くのマイコンショップまたは、エヌ・テイ・エル産業㈱宛に品名と機種、氏名、住所、電話番号を記入の上、現金書留にてお送りください。(送料は無料サービス)

制作 株式会社チャンピオンソフト
〒530 大阪市北区西天満6丁目1番12号瑞穂ビル
TEL(06)365－9900
ファックス(06)361－1176

発売元 エヌ・テイ・エル産業株式会社
〒657 神戸市灘区篠原北町1丁目5番13号
TEL(078)881－1914
ファックス(078)882－0555

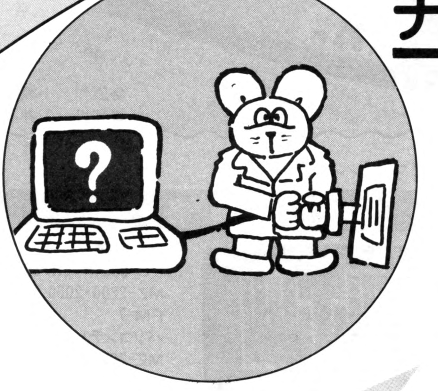
ユニークなオリジナルソフトで躍進する

CS CHAMPION SOFT

ソフト界のスターへの登龍門

株式会社チャンピオンソフトでは、マイコンユーザーの優れたソフトを商品化いたします

チャンピオンソフトデザイナー募集



さあ、君もソフト界のスター・チャンピオンソフトデザイナーとしてデビューしよう!

募集内容

①ゲームソフト(反射神経型、シミュレーション型、知的戦略型、その他なんでも結構です)
②実用ソフト
③教育用ソフト
※応募はオリジナルで未発表のものに限ります。他人の版権を侵害するもの、単に他機種に移植したものはお断りいたします。
※現在開発中のもの、開発計画段階のものについてもご一報ください。

使用機種

①NEC……PC-8001MKII, PC-8001, PC-8801
②SHARP…MZ-2200, MZ-2000, MZ-700, パソコンテレビX1
③富士通……FM-7/8
④松　下……JR-200

メディア

今回はテープベースのものに限定いたします。

応募方法(期限はありません)

下記事項を明記の上、プログラムリスト及びプログラムをSAVEしたカセットを添付して㈱チャンピオンソフト開発室宛お送りください。
①住所 ②氏名 ③年令 ④職業(勤務先又は学校名) ⑤電話番号 ⑥生年月日 ⑦マイコン歴 ⑧プログラム名 ⑨使用機種と必要周辺機器 ⑩使用言語 ⑪必要メモリ ⑫プログラムの概要 ⑬プログラムの操作手順
※応募いただいた原稿、リスト、カセットは原則として返却いたしません。

審　査

応募いただいたソフトは、その都度審査を行い採用作品は、ご本人に連絡するとともに速やかに商品化いたします。

版権及び著作権使用料

採用作品の権利一切は弊社に帰属いたしますが著作権使用料として下記のものをお支払いいたします。
①商品化決定時に…10,000円, 20,000円, 30,000円
(ソフトの内容による)
②発売後売上本数1本につき…………定価の10%
(例) 定価3,500円のソフトが
●300本売れたら30,000円+350円×300本=135,000円
●500　〃　30,000円+350円×500本=205,000円
●1000　〃　30,000円+350円×1,000本=380,000円
他機種に移植した場合は、原則として原作者½、移植者½

ソフト開発要員募集

とにかくマイコンが好きな人。

当社であなたの能力をフルに発揮してみませんか。

①職　種:ゲームソフト・実用ソフト・教育ソフトの開発
②資　格:何よりもマイコンが好きな人(自宅通勤可能な方)
③給　与:能力給を基本とします。
④勤務地:大阪
⑤勤務時間:午前9時~午後6時
*(株)チャンピオンソフト宛履歴書(写真貼付)を郵送下さい。面接日を通知します。
※応募の秘密は厳守します。

チャンピオンソフト アルバイター大募集

学生アルバイト・サラリーマンのサイドビジネスに最適。何によりもマイコンが好きな人。余暇を利用し、あなたの優れた能力をチャンピオンソフトで役立ててください。

アルバイトの登録制度を設けています。ご希望の方は、①氏名 ②住所 ③TEL ④職業(勤務先又は学校名) ⑤年令 ⑥所有マイコン機種(周辺機器も併記してください)を記載の上(株)チャンピオンソフト宛お申し込みください。ご遠方の方でも結構です。
*マシン語, コンピュータグラフィックに詳しい方特に歓迎!!

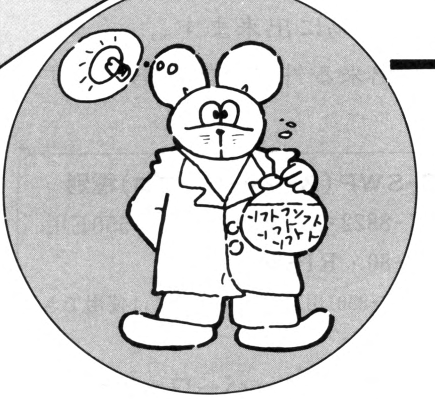
CS 株式会社チャンピオンソフト
〒530 大阪市北区西天満6丁目1番12号 瑞穂ビル
T E L (06) 365－9900
ファックス (06) 361－1176

ユニークなオリジナルソフトで躍進する

CHAMPION SOFT

アイデアマン全員集合！

株式会社チャンピオンソフトでは、マイコンユーザーの優れたアイデアを商品化いたします



アイデアマン全員集合

最近のマイコン市場の急速な拡大に伴ない、マイコン用ソフトの需要も急速に拡大してきておりますが、マイコン用ソフトに対するニーズは、マイコンユーザーの皆様が一番よくご存知であり時間的制約等により、自分自身でソフトの開発はできなくても、**こんなソフトを開発できたら**……というアイデアはたくさんおもちのことと思います。当社では、このようなアイデアを組織的に集める機関として**アイデアバンク**を設置いたしました。

マイコン用ソフトに関するアイデアをお持ちでしたら是非下記要領により当社「アイデアバンク」に登録してください。ご登録いただいたアイデアは、審査の上採用分につきましては、規定のアイデア料をお支払いするとともに、当社開発室のシナリオ担当、デザイン担当、音楽担当、プログラマーを動員して商品化し、ソフト流通商社株式会社日本ソフトバンクを通じて、他のユーザーの皆様に提供していく予定です。ふるってご応募ください。

アイデアバンク登録要領

募集アイデアの範囲

Ⓐゲームソフト関係
Ⓑ実用ソフト関係
Ⓒ教育ソフト関係
※応募は、オリジナルで未発表のものに限ります。
　他人の版権を侵害するものはお断りいたします。

応募方法（応募期限は無期限です）

下記事項を明記の上、(株)チャンピオンソフト アイデアバンク宛お送りください。
①住所　②氏名　③年令　④職業
⑤電話番号　⑥マイコン歴　⑦マイコン所有機種
⑧アイデアの区分(ゲーム、実用、教育)
⑨アイデアの名称
⑩概要・画面スケッチ・キャラクター・ゲームの場合ルールなど(A４サイズ用紙に５枚程度)

審　査

ご応募いただいたアイデアは、アイデアバンクに登録するとともに、採用のものは、ご本人宛ご連絡いたします。

アイデア料

採用アイデアの権利一切は、弊社に帰属しますが、アイデア使用料として下記のものをお支払いいたします。
①チャンピオン賞……………………100,000円
②優　　秀　　賞……………………50,000円
③金　　　　　賞……………………30,000円
④銀　　　　　賞……………………20,000円
⑤銅　　　　　賞……………………10,000円
上記の外売上本数１本につき定価の１％をアイデア使用料としてお支払いいたします。

株式会社チャンピオンソフト
〒530 大阪市北区西天満6丁目1番12号 瑞穂ビル
ＴＥＬ(06)365－9900
ファックス(06)361－1176

RP-80・FP-80対応

シリアルドットプリンタ用

PC-8801 日本語ワードプロセッサ

PC-SWP

¥39,800

▶20,000語辞書が標準装備され、なおかつユーザーが登録することが出来ます。
▶漢字への変換方式はローマ字・カナ・16進の3種類の入力が出来ます。
▶罫線は実線・細線・点線・アンダーラインの4種があり、作表が容易に出来ます。
▶240件の任意の文字または記号などをユーザーが作成することが出来る外字機能があります。
▶CRTは200ライン及び400ラインどちらでも使用可能です。
▶横方向への画面スクロールにより最大53桁まで文書の作成が出来ます。
▶文字の種類も倍幅・全角・½文字の3種類あり、また縦書き出力も出来ます。

PC-SWP(スーパーワープロ)種別
・PC-8822・NK3618-22・GP-550E用
・FP-80・RP-80用
注 PC-8801用MPシリーズでは使用できません。

PC-9800用スーパーワープロ
近日発売!

FM-7決定版 ワープロ

文節単位での変換処理が可能

FM-7専用 日本語ワードプロセッサ

FM-JWP/7

ワープロ本体	¥20,000
本体+タイプレッスン	¥25,000
タイプレッスン	¥ 6,500

▶漢字への変換方式にはローマ字・カナ・16進の3種類の入力ができ、また送り仮名付き入出力が可能となり、いわゆる文節単位での処理が出来ます。
▶20,000語辞書が標準装備されているほか、127パターンの外字登録機能があります。
▶1ドライブ及び2ドライブのどちらをお持ちの方でも、機種選択処理によりお使いになれます。
▶罫線は実線・細線・点線・アンダーラインの4種類があり、作表が容易に出来ます。
▶文字の種類には倍幅・全角・½文字の3種類があり、また縦書き出力も出来ます。
▶移動・複写などを含む豊富な編集機能があり、また簡単な計算処理も可能です。
▶本プログラムに別売オプション(¥19,800近日発売)を組み合せますとFM-JWP/7Sと同じものとなります。
※本プログラムはFM−8では使用出来ません。

・FM-JWP/7S ¥39,800
〈近日発売〉
FM-JWP/7のスペシャル版です。

・FM-PLAIN ¥20,000
漢字版作表簡易言語

ＳＢＣお手伝いさんシリーズ

お寺さんのお手伝い

（ＦＭ－７・８用檀家管理システム）
基本部 ￥150,000　オプション部 ￥50,000

● 1000軒の住所・氏名等を出力でき、回忌の案内及び明細、宛名印刷ができます。
● お布施、寄付金、年会費等の実績管理が出来ます。
● 過去台帳の登録、檀信徒の登録、弊社にて承ります。（有償）

酒屋さんのお手伝い

（PC 8801・PC 9801・FM－7・8・11 酒類小売業システム）
￥250,000

● 酒類小売販売店にとって最も繁雑な業務の税務署への報告書作成及び指定の酒類受払帳の作成等を自動的に行います。
● 出力帳表としては、仕入先別仕入月報・容量別在庫月報・品目別在庫月報・酒類別在庫月報・酒類受払帳等があります。

歯医者さんのお手伝い

（PC 9801・FM－11　予約管理・窓口会計業務）
￥200,000

● 患者さんの予約システム、時間単位、曜日単位、週単位、月単位（最大３ヶ月先迄）の予約が出来ます。
● 患者さんの検索システム、カルテNo. による予約月日時分の検索、氏名による検索、月日時分による検索等及び予約票の発行が出来ます。
● 窓口会計業務システム、保険点数入力による患者一部負担金の割出、領収書の発行（レジスタ機能）カルテNo. 氏名の領収書への表示等を行います。
● 患者別、保険種類別の日計表・月計表・年計表等を作成する事が出来ます。
● バージョンアップにともなう継続的保守も行います。

見積書作成のお手伝い

（FM－7・8・PC 8801・PC 9801 塗装面積計算システム）
￥250,000

● 計算式はどんな式でも最大32式まで自由に登録、修正する事が出来ます。
● 変数項目は最大32項目まで使用出来ます。
● 集計機能があり、各識別記号により、小・中・大計をとる事が出来ます。
● 計算書のタイトル名を付けられ、全て会話形式で作業を進められますので操作が簡単です。
● 本システムは橋梁設計で塗装面積の見積用として開発されたものですが、あらゆる見積作業に対応出来るよう、システムの変更等承っております。御気軽に御相談下さい。

漢字プログラム作成のお手伝い

（PC 9801　漢字・熟語サブルーチン）
各￥20,000

貴方のプログラムを漢字システムにしてみませんか…………。

● みなさまの開発されたプログラムにサブルーチンの組込だけで漢字仕様のプログラムに変身します。
● 内容としましては漢字・熟語の画面及びプリンターへの入出力ルーチン、漢字データのディスクへの格納方法。ユーザープログラムとサブルーチンとのデータ受け渡し方法等の使用方法がマニアル及びサンプルプログラムで説明されておりますので、初心者の方でも簡単に取り扱えます。
● 漢字・熟語とも登録、削除が自由に出来、自分本意の辞書作成が出来、巾広い用途にお応え出来るものです。
● プログラミングは感性、そのアイディアがパソコンを救う………どんどん発展させて下さい。尚、弊社ではサブルーチンを利用したみなさまのソフトを御相談の上買い上げ、責任をもって販売も致しております。どしどし、御応募下さい。

財務会計システム

FM-ZAIMU

（FM 7・8・11用DISK版 財務管理システム）三和ビジネス㈱製
・ビジネスプリンター・シリアルドットプリンター
基本部 ￥150,000
オプション部 ￥50,000

● ワープロ機能を備えた本格的漢字版財務会計システムです。
● 勘定科目最大240、補助科目は任意に拡張でき、1枚のフロッピーに10,000件の仕訳が入力保存できます。

PCA高級財務会計システム

（MZシリーズ・FM－7・8・11 PC 9800）ピーシーエー㈱製
￥98,000　￥148,000　￥198,000

● 会計専門家数名のノウハウを結集したシステムです。
● 日常良く使われる主要48科目は、ワンタッチ入力方式になっています。
● 入出金データーの入力は自動仕訳方式にする等、使いやすく、正確で、しかも主要部分は機械語処理により、トータル・パフォーマンスを追求しています。
● 主要帳票はオフコン並の専用用紙が用意されておりますので、銀行や税務署等にそのまま提出できます。

パソコン・
プログラマー
大募集中!!

株式会社 エス・ビー・シー

〒231 横浜市中区本町５丁目49番地
丸忠センタービル３Ｆ
TEL045-201-８４８０（代）

あなたのパソコンを
うつくしくよそおう

キーボードカバー

●美しくキーボードをカバーします
●デモ中に他の人の手が触れるのをガードします
●アクリル製

富士通パソコン用

FM-7
FM-8
FM-11

(各)¥3,800

●カラー：ブラウンスモーク

KEYBOARD COVER

イニシャルプレゼント

キーボードカバーお買いあげの方にもれなくイニシャルシートをプレゼントいたします

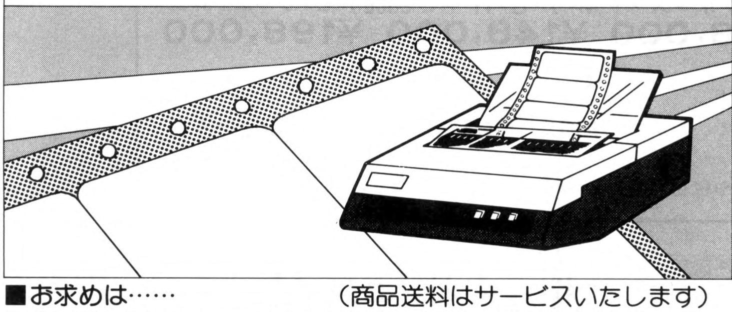
他に シャープ X1・MZ2200・MZ2000、NEC PC6001・8001・8001MKII・8801・9801、東芝 パソピア7、日立 ベーシックマスターL3(M2、M5)用もあります。(各)¥3,800

マイコン用

宛名 タックシール

●4インチ幅トラクタフィールド用紙 (型番 TC-10)
●1ケース・1ロール500枚入×2コ入
●シールサイズ(タテ)36mm×(ヨコ)86mm

¥3,200

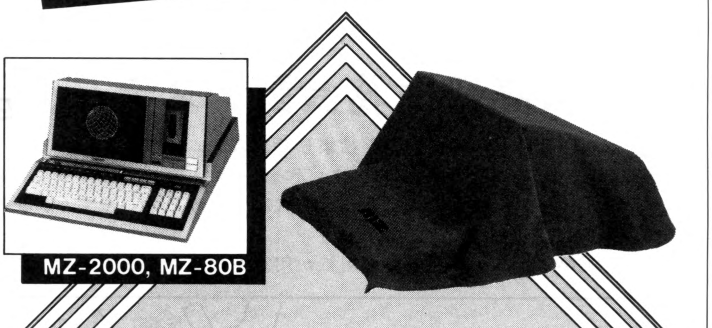


シャープMZ用

●あなたのマイコンをドレスUP!

マイコンカバー

¥3,000 ●材質：綿 100% ●カラー：エンジ・紺



MZ-2000, MZ-80B

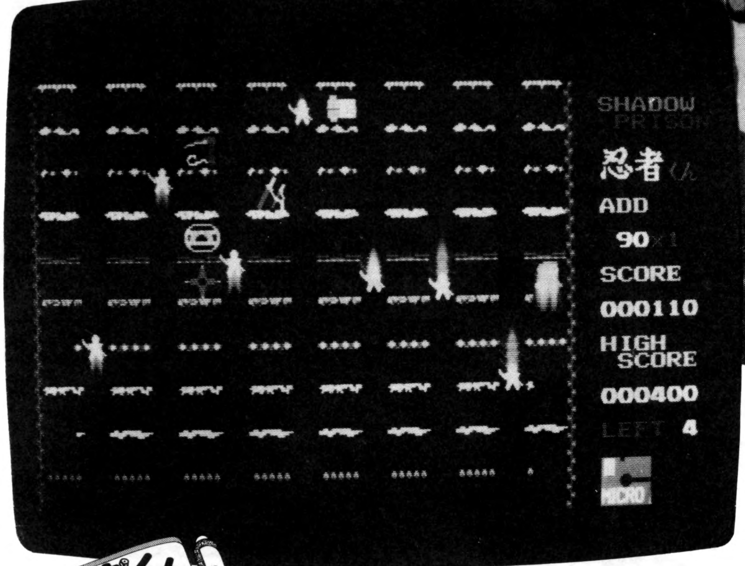
MZ-1200にも掛けられます

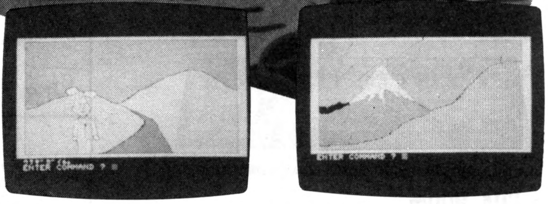
■お求めは…… (商品送料はサービスいたします)
全国パソコンショップまたは直接右記へお申し込みください。
● 現金書留でご注文の場合は商品名、住所、氏名、電話番号をはっきり書いてお送りください。
● 銀行振込みでご注文の場合は商品名、住所、氏名、電話番号をハガキまたは電話でご連絡ください。

総発売元 エイト電気株式会社
〒110 東京都台東区上野5-3-4 ☎03-831-5632(代)
振込銀行：北陸銀行 上野支店 (普)No.4032110

西日本発売元 三栄電子株式会社
〒556 大阪市浪速区日本橋東2-10-2 ☎06-643-3833(代)
振込銀行：三和銀行 恵美須支店 (普)No.189879

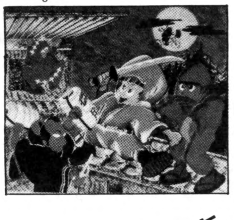
どちらも、超！自信作です。





忍者くん

忍者くん



とても可愛い忍者くん、どうしても最上階へ行かねばならない。
しかし、七人の恐しい役人が、すきあらばと、忍者くんを狙っているのだ。斬られずに無事、最上階へ到達できるかどうかは、君の腕次第！ プレイしている人も、見てるだけの人も、思わず声を上げてしまう程、実におもしろくスリリングなゲームの登場だ。

新発売 **PC-8801(C)　¥3,800**
PC-8801(FD)　¥5,800
開発中 **X-1(C)**

ドリームランド

君は眠りについた——いつのまにか、夢の中とも知らず、さまよい歩く冒険者になり変わっている。様々なグラフィックが織りなす、ファンタジックで奇妙な、不思議な世界にいる。君の冒険心はさらにかきたてられるのだ。この幻想の夢から、果たしていつ覚めるのか？　かつてないスケールのビッグアドベンチャーゲーム。画面イラストも60を超える超大作。君のアドベンチャースピリットにロマンが加わる。

新発売 **MZ-80B　MZ-2000(FD)**　2枚組
開発中 **PC-8801　PC-9801(FD)**
¥12,800

この夏、話題の新作ソフトずら〜り勢ぞろい!!

ピコピコ

MZ-2000
X-1
¥3,800(C)

36個のブロックを、攻撃、防御に使ったり、自ら発射する弾丸で敵を攻撃。死体は消えずに残るのでゲームが進むにつれ、複雑化してくる。敵はたまにワープするので御用心。

ダイヤモンドアドベンチャー

PC-6001
¥3,500(C)
X-1
¥3,800(C)

三階建てのビルに隠された、時価10,000,000円のダイヤモンドを探し出す。デジタルロックのナンバーがキーポイント！しかし、ビルの中はトリックだらけ、君を陥れるいくつもの罠が待ちうけています。

ミッドナイトコマンダー

PC-8801
FM-7
¥3,800(C)

真夜中の洋上、月あかりもなく、レーダーを頼りに敵を探索し、攻撃する。ちょっとしたミスから敵のレーダー網にかかり攻撃を受けたらたちどころに撃沈されてしまう。味方の艦隊、飛行機隊を駆使し、敵の本拠地を乗っ取れ！

エキサイティングゴルフ

FM-7
¥3,500(C)

ゴルフは、もう芝の上でするだけのものではなくなった!!グラフィックに展開されるコースは、もう君だけのもの！君の緻密な計算から打たれたボールはグリーンを捕らえることができるか？臨場感あふれる本格的なゴルフシミュレーションゲーム。

マイクロキャビン
販売／㈱ MICRO CABIN
企画・制作／㈲ ARROWSOFT
〒510 三重県四日市市鵜の森 1-2-15　メゾンヴァンベール2F　☎0593-51-6482

ドキドキサマー ワクワクマイコン

サマーセール実施中!!

夢のジョイント価格

(すべてカラーディスプレイ〈14インチ〉・データレコーダ付)

NEC-PC-8001マークII

合計標価 207,400円
サマー特価 ➡ 182,000円
月々 4,900円

富士通FM-7

合計標価 210,400円
サマー特価
⬇
185,00円
月々
5,000円

SHARPパソコンテレビX1

テレビ画像とのクロスオーバー
コンピューターグラフィックス

{ローズレッド
メタリックシルバー
スノーホワイト}

スーパーインポーズ機能 サウンド
ジェネレーター 他、盛沢山

「見る」・「創る」
「学ぶ」・「遊ぶ」

- CZ-800C/D
- グラフィックRAMCZ-8GR
- ソフト5本サービス

合計標価 300,000円
サマー特価
➡ 260,000円

頭金0円・ボーナス0円・48回均等払い

東芝パソピア 7

合計標価 204,200円
サマー特価 ➡ 172,000円 月々 4,600円

日立ベーシックマスターMARK5

合計標価 202,400円
サマー特価 ➡ 172,000円
月々 4,600円

カシオFP-1100

合計標価 212,400円
サマー特価 ➡ 152,000円
月々 4,100円

QC-10 日本語パーソナルコンピュータ

(CPU＋2ドライブ＋漢字)

日本語の、日本語による、日本語のためのスーパーパソコン●メインRAM256KB実装●640×400ドット●日本語CP/M、日本語スーパーカルク他、圧倒的強力ソフト群サポート●専用機なみ日本語ワープロ機能●データ通信他、オールランドプレーヤー●自分の故障は自己診断(国内で初めて)

- ■QC-10(CPU＋ディスク320KB×2基＋漢字ROM) ■キーボード(ケーブル込み)
- ■専用グリーンCRT(ケーブル込み)

合計標価 495,000円➡ ウルトラプライス

専用カラー、専用漢字プリンター組合わせはお問合わせ下さい。(日本一安いクレジットは、ウルトラプライスより起算)

オフコンにせまる超本格派FM-11

16ビット・8ビット両用パソコン

- FM-11EX(MB25050)
- 増設ドライブ(MB27609)
- 専用カラー(MB27311)
- 専用プリンター(MB27403)
- システムソフト(CP/M-86, F-BASIC, 簡易言語FMCALC)
- プリンター用紙1,000枚とディスケット10枚サービス

合計標価 799,400円
➡ ウルトラプライス

ワープロコース

(漢字プリンター不要)

- FM-7・RGBVISION II
- カラーディスプレイ用ケーブル
- 漢字ROMカード
- TF-20・ミニフロッピー1/Fカード
- システムディスク
- RP-80・プリンター用ケーブル
- プリンター用紙(1,000枚)
- COMAS-WPV1・ディスケット(10枚)

合計標価 585,600円
特別価格 ➡ 495,000円

'83夏のビジネスフェア —見て・さわって・確かめる—

参加された方にもれなく粗品進呈!

7/22➡24	7/29➡31	8/5➡7	8/19➡21	8/26➡28	9/2➡4
エプソン QC-10 日本語パーソナルコンピュータ HC-20 持ち歩き自由型	パナファコム C-280M. 1台5役の他機能パソコン 簡易言語	富士通 マイオアシス2 ワープロも愉快な文房具 ワープロ	内田洋行 カマラード 1台で4役パワー全開 オフィスをシェイプアップ ワープロ	リコー リポート300 4通りの入力方式が選べる コンパクトワープロ ワープロ	オリベッティ M-20 10ビットCPU 搭載の プロパソコン 戦略会計システム 英文ワープロ

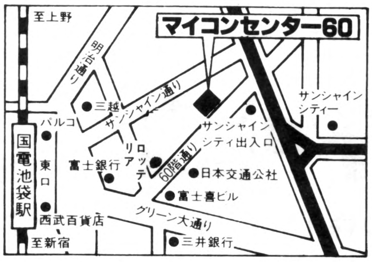
お電話1本、お葉書1枚で! 全国即納システム! (配送料は無料) あくまで良心的な営業サービス方針は熱烈大好評!!

お申込み方法

①商品名(メーカー、型番) ②合計金額(定価合計でも自動値引きされますからご安心ください) ③お支払方法(クレジット、現金、またはリース) ④クレジットの場合…月払い及びボーナス払いご希望額(ボーナス払いを多少なり入れますと月額は低くなります) ⑤お名前(20歳未満の方は保護者のお名前も)⑥年齢⑦ご住所⑧電話番号⑨職業⑩他、ご意見、ご希望以上を官製葉書にご記入の上、右記宛ご郵送ください。

※お金のやりとりは1円なしでも、お手元に全品が揃います。(頭金0円のとき)

●銀行振込ご利用の方は住友信託銀行池袋支店・普通口座No. 2706052にお振込みください。[口座名:(株)マイコンセンター 送金手数料は差引きもOK]



日本一安いクレジット!
1回から60回までの自由自在。
マイコンの新名所 池袋・東口。

—ご来店、熱烈歓迎!朝10時～夜8時—

お問い合せ・お申し込みは

マイコンセンター60

— oh/FM係 —

☎03-980-1360(代)

〒170 東京都豊島区東池袋1-21-5
サンシャインシティー出入口前

誰にでも手軽に使える高級ソフトウェア

使いやすさとトータル・パフォーマンスを実現しました。

PCA高級財務会計システム

◆富士通FMシリーズ用◆

〈監修〉　公認会計士　吉村　成弘
ピーシーエー株式会社取締役

〈開発・販売〉　ピーシーエー株式会社
都築電産株式会社 TEL(03)433-2171

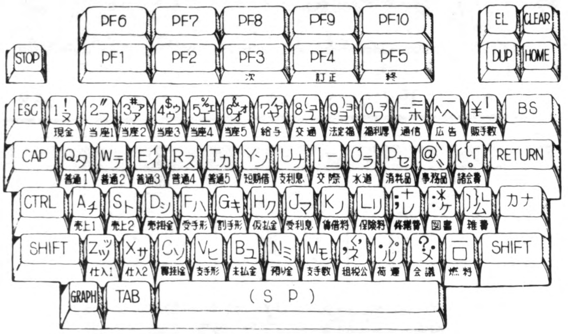
〈システムの価格〉 A(元帳、試算表、決算書)　¥ 98,000
B(A＋補助簿、手形管理)　¥148,000
C(B＋部門管理、その他分析)¥198,000

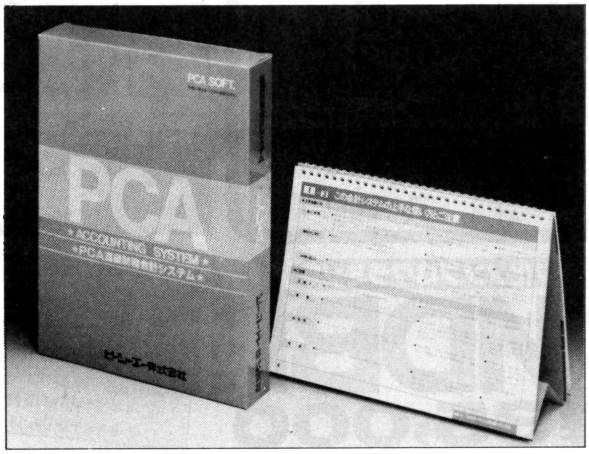
〈特　長〉
○あなたの会社に合わせた科目登録や補助簿登録ができ、合計金額1,000億円未満の会計処理ができます。
○主要帳票はオフコン並の専用用紙が用意されておりますので、銀行や税務署等にそのまま提出できます。
○主勘定科目最大240、補助科目最大240が用意され、補助科目の活用により、売掛金、買掛金、手形管理など広範囲な処理ができます。
○部門設定が9分類でそれぞれの損益計算書ができます。
○1ヶ月 2,000仕訳の処理ができます。
○入力方法は入金、出金、単一振替、複合振替が可能で、入出金は自動仕訳方式になっています。
○画面は漢字の大文字で表示され、非常に見やすくなっています。
○日常良く使われる主要48科目は、ワンタッチ入力方式になっています。(下記キーボード参照)
○帳票名を選択するだけで自動的に分類集計する等、使いやすく、正確で、しかも主要部分は機械語処理により、トータル・パフォーマンスを追求しています。
○処理範囲によりA.B.Cと3段階のシステムが用意され、必要なレベルを選択できます。

MENU 1

```
0: MENU 2へ            6: 科目・残高の登録・変更
1: データ 入力          7: 日 計 表
2: データ 検索・訂正    8: MENU 3へ
3: チェックリスト
4: 仕 訳 帳
5: 科目残高登録リスト
```

処理を 選択して下さい ?





《システムの概要》

	項目	システム A	システム B	システム C
日常処理	前準備処理	○	○	○
	データ入力作業	○	○	○
	データの各種検索，訂正	○	○	○
	入力データーチェックリスト	○	○	○
	仕訳日記帳	○	○	○
	日計表		○	○
任意処理	元帳	○	○	○
	補助元帳		○	○
	手形管理表		○	○
	補助科目残高一覧表		○	○
	合計残高試算表(B/S, P/L)	○	○	○
決算処理	決算報告書			
	貸借対照表，損益計算書	○	○	○
	販売管理費明細書	○	○	○
	製造原価報告書	○	○	○
	利益処分計算書	○	○	○
その他管理資料	取引先別総合収支明細表			○
	資金繰実績表			○
	経営分析表			○
	部門別損益計算書			○
	部門別P/L一覧表			○
	月次残高推移表			○
	三期連続P/L対比棒グラフ			○

※税務申告ソフト開発中！

専門家の設計による使いやすいビジネスソフト

ピーシーエー株式会社

東京本社　〒160 東京都新宿区西新宿7丁目7番33号
(新銘ビル新館2F) ☎03-368-9631(代)
大阪本社　〒530 大阪市北区中崎西2-2-1東梅田八千代ビル3F　☎06-315-8637
名古屋本社　〒451 名古屋市西区菊井2丁目6-7　☎052-565-0406

㈱日本ソフト&ハード社

Computer 11

コンピュータ・イレブンは 株日本ソフト&ハード社の直営ショップです

ソハコムはユニークな
イレブンオリジナル
ブランド商品です

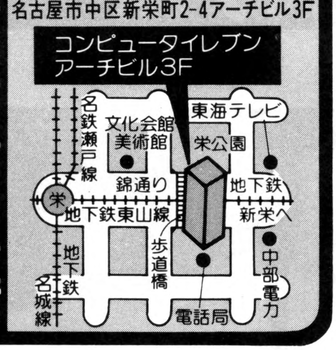
SOHACOM（ソハコム）

6月18日㊏名古屋駅前12号店 ビッグにオープン!! ☎052(971)3421㈹

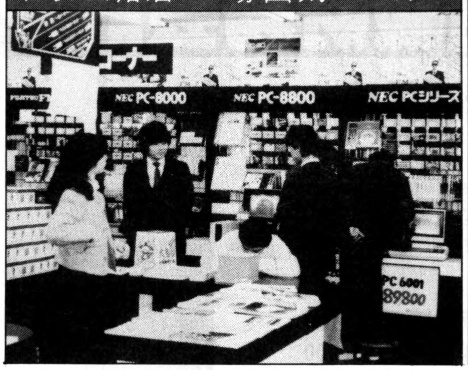
名古屋栄町12号店

☎052(971)3421

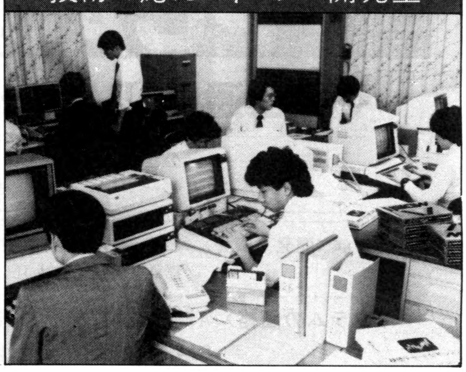
名古屋市中区新栄町2-4アーチビル3F



ゆったり落着いた雰囲気のショップ



技術で応えるイレブン開発室



グラフ化ノンプログラミング

MDB11

SOHACOM

¥35,000

（漢字対応型〈FM8用〉は1万円高となります）

メーカー認定ソフト
MDB11（グラフ化処理機能付）
if800 PC-8001 PC-8801
PC-9801 FM-7 レベル3
FUJITSU MICRO8 MULTI16 用
（300ページ完全解説書付）
5インチ・8インチ版もあります。

簡易言語MDB11の7大特徴

❶幅広いニーズにお応えします。——
各種のカード分析（顧客カード・会員カード・人事カード）、文献検索、実験整理、インデックス・コード表作成、ダイレクトメール、現金出納帳……。

❷データの有効活用のキメテです。
データベースですから、一つのデータを使っていろいろな仕事ができます。同じようなデータを重複して打ち込んでいた無駄がなくなります。

❸データは1,000件から最大30,000件まで記録。（インデックスをつけない場合）
しかも、1,000件のデータから1件を探すのに5秒。高速索引検索機能付き。

❹もちろん分類・並べ換えは自由。
カラーグラフ化も一発です。（棒グラフ・折線グラフ・円グラフ）

❺簡易言語の中に簡易言語。——
強力な報告書（レポート）作成機能付き。

❻システムのソフト内容はユーザーに完全公開。
使い方やリストを詳説した、340ページ（B5版）のマニュアルがついています。改造も自由にできます。

❼MDB11は、「使えば使うほど味が出て、手放せなくなるソフト」

説明資料100頁無料

グラフ化OK!!万能データ管理プログラム

ソフトリスト全公開!!グラフ化処理機能付

SOHACOM

数値計算パッケージ

LET'S calculate!

FM7/8用・PC8001マークⅡ用
PC8801用

（テープ版¥4,800）

最大最小値・平均・標準偏差・ソート・行列の積・逆行列・
連立一次方程式の解・多項式の根・補間法（一次回帰）

●科学技術計算・統計計算の基本となる6つのプログラムを収録。
●対話形式でコンピュータ知識不要。●データを入力するだけで、ムダな計算時間を追放。簡単に計算結果が得られます。●最大最小値は500ヶまでのデータ処理が可、さらにデータの平均値行列の積は、要素数3,600ヶまで計算。●逆行列は40次まで求められます。●詳細マニュアル付

SOHACOM

パソコンでその日から使える

株価チャート分析システム

TYPEⅡ ¥100,000

●表示できるチャート。
ローソク足・移動平均・出来高・信用取引残高・カイ離度相対レシオ・新値三本足・カギ足等

●見てください。実績です。
2年前発売以来、証券会社、各社資金運用部、出版社などに1000本以上導入され「株のプロ」に愛用されています。

●カンピュータよりコンピュータ
13種のチャートによる総合判断が可能です。あらゆる角度から全体的な動きを把握できます。

●データ入力わずか4分。連続自動処理機能付。
64銘柄のデータは4分で入力できます。その後は計算・作図・表示・プリンタ出力は完全無人で自動実行してくれます。

●「自分だけの必勝法」に活用してください。
実際の株価を入力しながら、パソコンと対話していると極めて有効なチャートを発見できます。

1000本突破!!

SOHACOM

イレブンオリジナル 日本語ワープロソフト

thePC漢字（PC8801 PC8001mkⅡ用） **theFM漢字**（FM7 FM8用）

●エプソンプリンター用 ¥4,800（カセットテープ版） ¥6,800（ディスケット版）

●EPSON MP80・RP80・FP80プリンターが日本語ワープロになった。
●オールマシン語の日本語ワープロ。
●漢字はカーソルによりスクリーンで編集。

漢字プリンターに変身!!

SOHACOM

イレブンオリジナル・ゲームソフト

江戸時代の名人 **本因坊秀策「囲碁トレーナ」**（PC8801用）

●秀策の碁譜は黒番のバイブル。大局観の養成にもってこい。自分の打碁の記録もできます。（ディスク版）…… ¥8,800

(10時〜7時営業) パソコンショップ

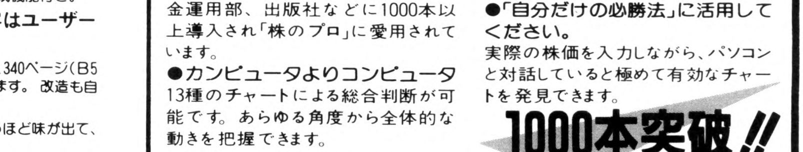
Computer 11

高田馬場駅前店	東京都新宿区高田馬場2-17-4	菊月ビル3F	〒160	☎(03) 209-7376㈹
池袋西口店	東京都豊島区池袋2-13-1	岸野ビル3F	〒171	☎(03) 980-1271㈹
新宿西口店	東京都新宿区西新宿1-9-13	高倉第2ビル1F	〒160	☎(03) 342-4821㈹
新橋東口店	東京都港区東新橋1-1	渡部ビルB1	〒105	☎(03) 572-5166㈹
上野駅前店	東京都台東区東上野2-19-3	第2吉沢ビル2F	〒110	☎(03) 833-0341㈹
横浜西口店	横浜市西区南幸2-5-4	深沢ビル1F	〒220	☎(045) 312-4611㈹
名古屋駅前店	名古屋市中村区椿町1-16	リクルート名古屋ビル5F	〒453	☎(052) 451-7371㈹
大阪難波店	大阪市南区難波2-1-2	太陽生命難波ビル3F	〒542	☎(06) 213-7387㈹
神戸三宮店	神戸市中央区三宮町2-1-5	センタープラザ西館3F	〒650	☎(078) 332-3961㈹
大阪ニュー梅田店	大阪市北区堂島2-2-2	近鉄堂島ビル7F	〒530	☎(06) 346-1552㈹
名古屋栄町店	名古屋市中区新栄町2-4	アーチビル3F	〒460	☎(052) 971-3421㈹

振込先

東京	三菱銀行高田馬場支店 ㊟4529956
池袋	武蔵野銀行池袋支店 ㊟032433
新宿	富士銀行新宿西口支店 ㊟1558871
上野	三井銀行上野支店 ㊟5076312
横浜	平和相互銀行横浜西口支店 ㊟0306345
名古屋	東海銀行笹島支店 ㊟188805
難波	第一勧業銀行難波支店 ㊟1003952
新橋	大和銀行新橋支店 ㊟6035336
神戸	第一勧銀神戸三宮支店 ㊟1186059
新梅田	太陽神戸大阪駅前支店 ㊟3124146
名古屋栄	東海銀行栄町支店 ㊟167834

東京高田馬場店	東京池袋西口店	新宿西口店	新橋東口店	上野駅前店	横浜西口店
☎(03)209-7376㈹	☎(03)980-1271㈹	☎(03)342-4821㈹	☎(03)572-5166㈹	☎(03)833-0341㈹	☎(045)312-4611㈹



名古屋駅前店	大阪難波店	神戸三宮店	大阪ニュー梅田店	名古屋栄町店
☎(052)451-7371㈹	☎(06)213-7387㈹	☎(078)332-3961㈹	☎(06)346-1552㈹	☎(052)971-3421㈹

イレブンDAY▶毎月11・12・13日（超特価日）

クレジットカードOK!!

官公庁・学校・企業の方には所定様式で迅速にお見積り致します。（4年・5年・6年の簡易即決リース有）

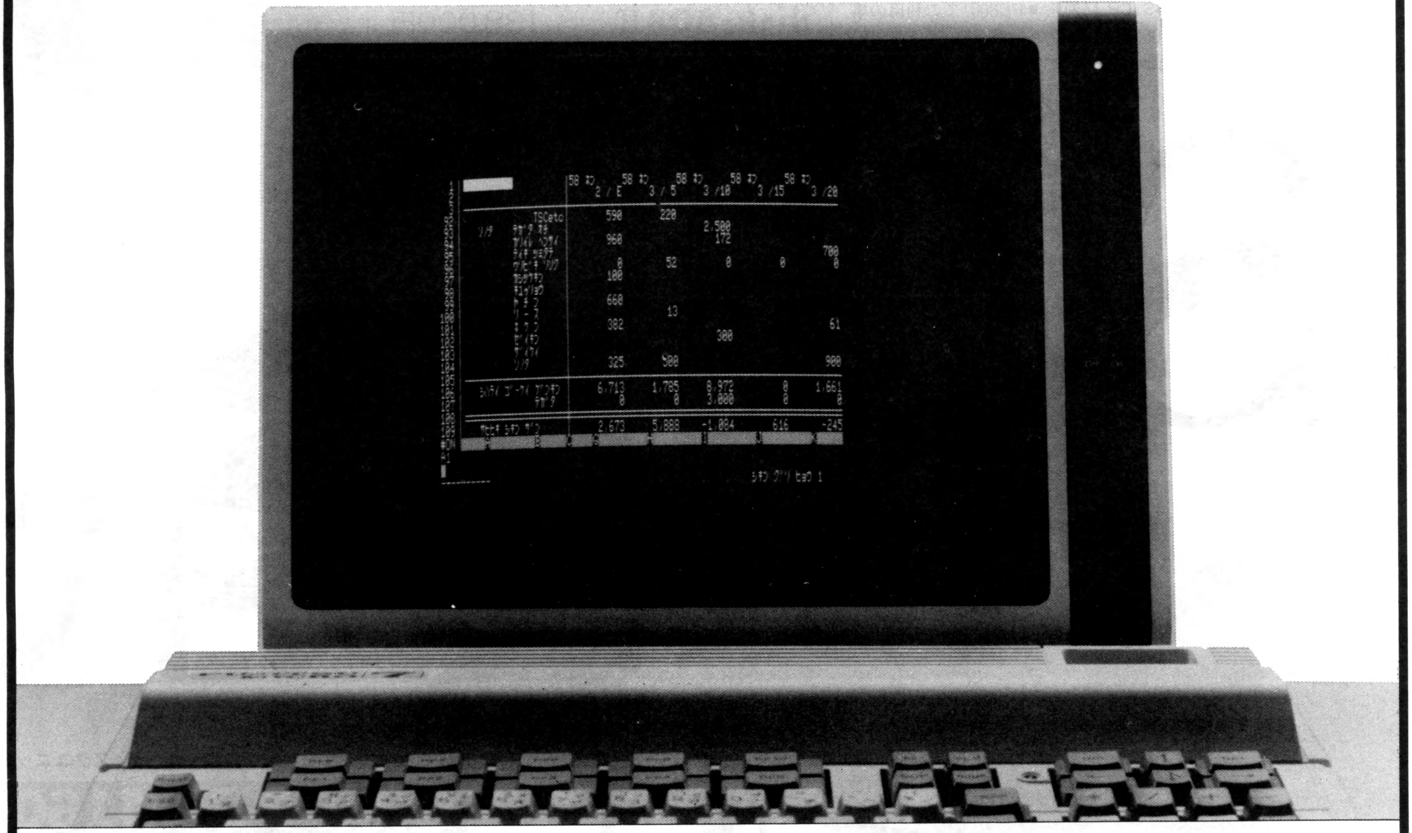
資料請求券 ソハコム OhF-7

㈱日本ソフト&ハード社は教育・開発・出版・卸・小売・通販のパソコン総合企業です。

〒171 東京都豊島区高田3-11-14 藤間ビル ☎03(232)0541㈹

作表無限、これは使える。

即実戦力。ビジネス経営に威力を発揮するFMCALCです。



ソフトマート簡易言語アプリケーション

FMCALC

FUJITSU MICRO 7 8用

ディスクバージョン……………………¥68,000
カセットバージョン……………………¥15,000

簡潔な命令で必要十分な処理を!

ワークシート方式で集計・ソート(アイウエオ順等の分類・並べかえ)などのわずらわしい業務を簡単なオペレーションで処理することができるようになります。FM-7、FM-8のハードウェアを効率的に使用することと、できるだけ少ない命令語で目的の処理を行なうことに工夫がこらされています。大企業の各セクションや、中小企業の経営等に威力を発揮する高速・高能力ビジネスソフトです。

主な特長

- ●POWER ONでFMCALCが起動します。他のソフト(FLEX、BASICなど)は不必要です。
- ●たいへん見やすい画面構成になっています。メッセージもできるだけ日本語です。
- ●数値精度は16桁、関数精度は13桁と、高精度の試算を高速で実行します。
- ●FM-7、FM-8のハードウェアと6809CPUの特性をフルに使用した機械語で書かれています。

SMart
SOFT MART, inc.

TSC社日本総代理店・スーパーソフト社 JAPAN AUTHORIZED DISTRIBUTOR

ソフトマート株式会社

〒101 東京都千代田区神田須田町1-18-6 第1谷ビル ☎(03)-256-5881

■取扱店

INTERRUPT東京 ☎(03)256-6325
INTERRUPT横浜 ☎(045)312-2325
INTERRUPT大阪 ☎(06)245-7575
INTERRUPT福岡 ☎(092)671-2466

5"デュアルディスクユニット

快走！disk-80markII

大好評のdisk-80markIIシリーズ。
高速に加え、さらに静寂・高性能。独走中。

FM-7／FM-8専用
disk-80FII　128,000円

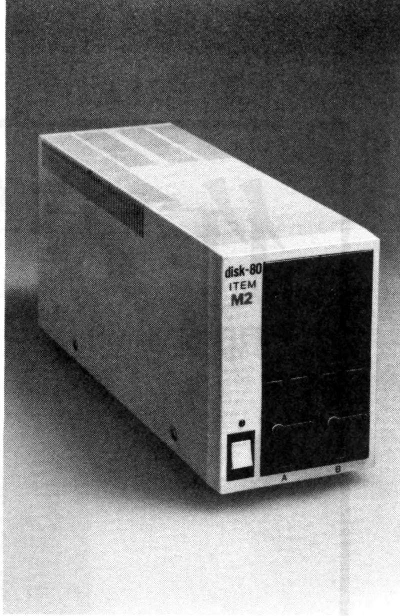
■DISK BASICをはじめFLEXなど各DOSに対応。
■FM-7の高速バージョンの対応も可能です。■接続にはdisk-I/O F(15,000円)とCBL-3(5,000円)をご利用下さい。

PC8801／PC9801／PC8001markII／PC8001専用
disk-80PII　123,000円

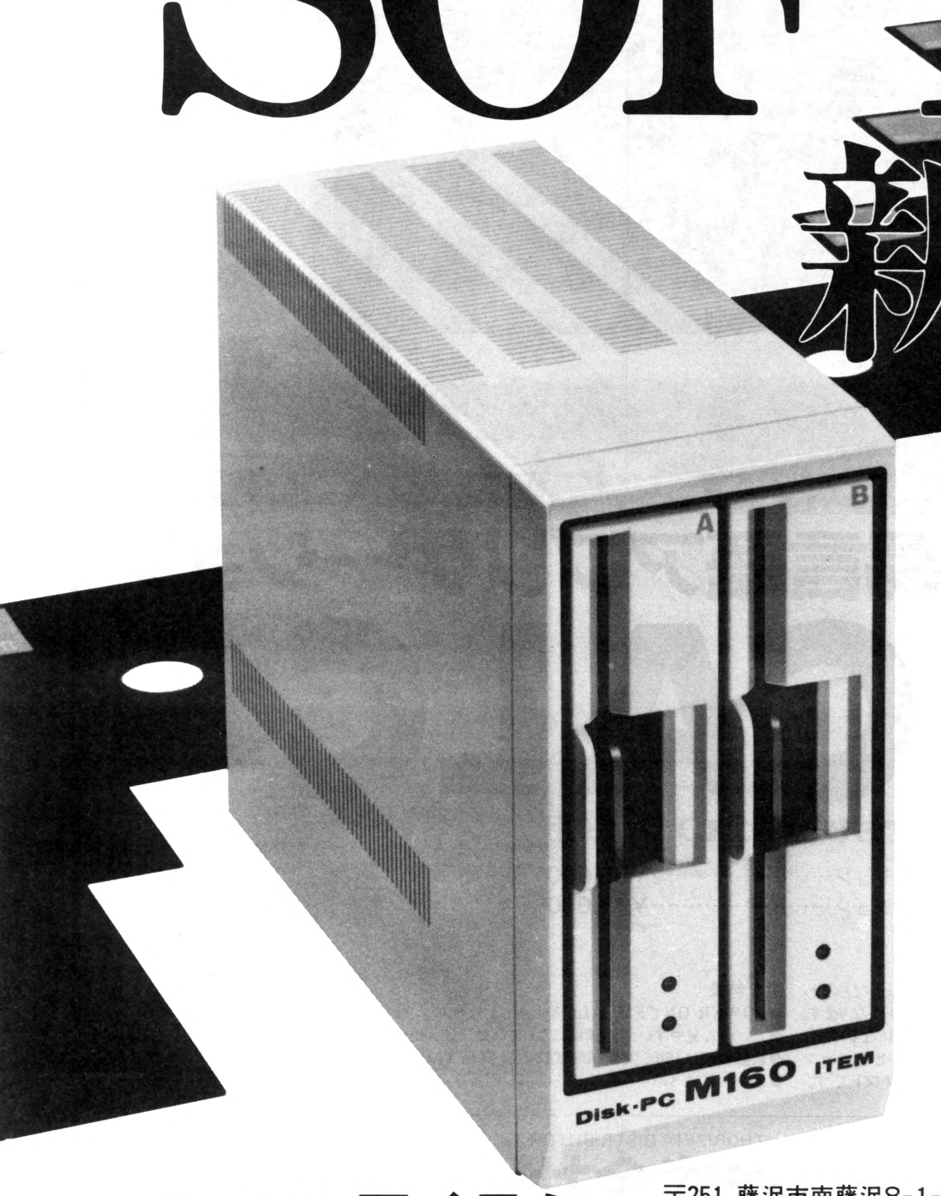
■RACET NEC DOS、CP/M対応 ■PC8001でご利用の場合はdisk-I/O P(12,000円)と接続ケーブルCBL-1(5,000円)をご利用下さい。他はCBL-2(5,000円)でダイレクト接続。システムディスケットは各対応のものをご利用下さい。

MZ-80B／MZ-2000専用
disk-80BII　138,000円

■フロッピーインターフェイス内蔵。拡張I/OポートがあればあとはIN接続ケーブル(CBL-3)でダイレクトIN。



SOFT新時代



高性能パワフル8インチフロッピーディスクユニット

Disk-PC M160

新発売　標準タイプ　256,000円

CP/M、MS DOS、P-SYSTEM…………
数々のパワフルなSOFTが続々登場してきます。まさにSOFT新時代。

スリムな薄型設計。大容量1MB×2ドライブ。
ビジネスに、パーソナルデータベースに、
ハイコストパァーフォーマンスを提供いたします。

モデル

M160-01(PC9801用)……………………256,000円
M160-02(PC9801用CP/M-86付)……298,000円
M160-03(PC8801用)……………………296,000円

※M160-02以外はシステムディスク(DISK BASIC等)が必要です。
接続ケーブル、インターフェイス等はふくまれています。

株式会社 アイテム

〒251 藤沢市南藤沢8-1-202
TEL (0466)27-1668(代)

□お求めは全国マイコンショップへ。
□電話によるご注文もうけたまわっております。
0466-27-1668(AM9:00～PM8:00)
□詳しいお問合せは左記あてへお申込み下さい。
□disk専用クレジットが用意されています。

＊CP/M®はデジタル・リサーチ社の登録商標です。＊UCSD Pascal™はカリフォルニア大学理事会の登録商標です。＊FLEX®はTSC社の登録商標です。

プログラム自由時代！ FM-7 8 11用

BASICジェネレータ

```
100 REM  ｹｲｻﾝ ﾌﾟﾛｸﾞﾗﾑ  (MAIN)
110 CLS
120 DIM X(2)
130 PRINT"  A X ^ 2 + B X + C = 0   / ｶｲ"
140 INPUT " A , B , C = ";W1,W2,W3
150 GOSUB "CALC" (W1,W2,W3,X(*),W$)
160 PRINT W$
170 PRINT X(1),X(2)
180 END
```

＋

```
100 '*** Subroutine (S1,S2,S3,S(*),SR$) CALC
110 '
120 S0=S2*S2-4*S1*S3:S(1)=0:S(2)=0:SR$=""
130 IF S0<0 THEN SR$="ｶｲ ﾅｼ":GOTO 170
140 IF S0=0 THEN SR$="ｼﾞｭｳ ｺﾝ"
150 S(1)=(-S2+SQR(S0))/(2*S1)
160 S(2)=(-S2-SQR(S0))/(2*S1)
170 RETURN
```

➡

```
100 REM  ｹｲｻﾝ ﾌﾟﾛｸﾞﾗﾑ  (MAIN)
110 CLS
120 DIM X(2)
130 PRINT"  A X ^ 2 + B X + C = 0   / ｶｲ"
140 INPUT " A , B , C = ";W1,W2,W3
150 GOSUB 63000
160 PRINT W$
170 PRINT X(1),X(2)
180 END
63000 '*** Subroutine(W1,W2,W3,X(*),W$) CALC
63110 '
63120 S0=W2*W2-4*W1*W3:X(1)=0:X(2)=0:W$=""
63130 IF S0<0 THEN W$="ｶｲ ﾅｼ":GOTO 63170
63140 IF S0=0 THEN W$="ｼﾞｭｳ ｺﾝ"
63150 X(1)=(-W2+SQR(S0))/(2*W1)
63160 X(2)=(-W2-SQR(S0))/(2*W1)
63170 RETURN
```

一度作ったサブルーチンをいつでも再利用できないか。既製のプログラムを新しいプログラムの一部に組み込めないか。—そんな要望に応えて、より簡単に、自由にプログラムが組めるソフトを開発しました。それがBASICジェネレータです。
プログラムを構造的に作る場合も、このソフトは威力を発揮します。文番号の自動置き替え、サブルーチン間の受け渡し、変数の自動置き替えにより、複数のプログラムの連結を行います。
付録としてデータの入力サブルーチンと、グラフ表示サブルーチンを数本用意してありますので、これでジェネレータの扱い方をマスターしてください。あなた自身のサブルーチンもディスクにSAVEしてジェネレータで利用できますので、自分独自のシステムを容易に再生することができます。

価格 **18,000円**(5Dディスク・マニュアル付)
当社では、分野別にサブルーチンを多数用意しています。
(有料¥3,000～　　　　／本)
BASICジェネレーター＋プログラマーズ"パル"
セット価格 **32,000円**

プログラマーズ"パル"

"パル"は、BASIC言語で作成されたプログラムを、コロン(：)、FOR-NEXT、IF THEN(ELSE)の段落にきって表示します。さらに指定によってプログラム中の使用変数、使用サブルーチンとび先の一覧を作成します。
"パル"によって、プログラムのデバッグを効率よく行うことができます。

主な機能

①1.構造化リスト 2.強化構造化リスト 3.変数番号対応表 4.サブルーチン文番号対応表（クロスリファレンスリスト）
②使用ファイル　非アスキーセーブ(通常のセーブ)のプログラム
③マシン語により高速処理実現
④変数文番号対応表、サブルーチン対応表では、特定の変数サブルーチンを指定することが可能。
⑤分割方式によるため、変数名の個数に制限はない。

```
100     REM          ======= TEST <<  DDDD
110     WIDTH 80 , 25
   :    LOCATE 6 , 2
   :    PRINT " TEST PROGRAM  " ;
   :    LOCATE 4 , 5
   :    PRINT "1. ｳｽﾞﾏｷ" ;
   :    LOCATE 4 , 7
   :    PRINT "2. ﾁｮｸｾﾝ ｶｲﾃﾝ" ;
   :    LOCATE 4 , 9
   :    PRINT "3. ｾｲﾎｳｹｲ ｶｲﾃﾝ" ;
120     LOCATE 4 , 16
   :    INPUT "SELECT NO. ==>>  " ; A$
   :    IF NOT (A$ = "1" OR A$ = "2" OR A$ =
        ELSE
          CLS
130     IF A$ = "1" THEN
            GO SUB 1000
   :        GO TO 110
        ELSE
          IF A$ = "2" THEN
              GO SUB 2000
   :          GO TO 110
          ELSE
            IF A$ = "3" THEN
                GO SUB 3000
   :            GO TO 110
140     A = 90
   :    FOR C = 0 TO 6
   :      FOR R = 0 TO 360
   :        X = 320+(C*36+R/10)*COS(R*3.1415
   :        Y = 99+(C*12+R/30)*SIN(R*3.14159
   :        PSET (X , Y , 7)
   :    NEXT R , C
```

価格 **18,000円**(ディスクバージョン)
価格 **15,000円**(テープバージョン)

ファイル入門

ランダムファイルの操作を解説付のプログラムで公開。あなたは明日からプロのプログラマー並みのファイル処理ができるようになります。簡易言語の不自由さを、独自のファイルを作ることにより解消。

■扱うファイル

①通常のランダム・ファイル
②インデックス付ランダム・ファイル
③グループ・インデックス付・ファイル

■扱う処理

①データの入力・修正・削除・プリント
②検索・ソート
③複数のファイルの結合　その他

価格 **18,000円**(5Dディスク・マニュアル付)

C.C.W PACKAGES

等高線作画 ¥100,000
XY一平面に分布する任意量測定又は計算結果の等高線(等値線)を描く。

グラフパッケージ ¥35,000
ファイルデータの受け渡しにより12種類のグラフを作成する。
カラー、パターンの選択によりプリンターにハードコピー可能

三次元座標基本ルーチン ¥45,000
2ベクトルのなす角、2平面の交線と交角、座標交換と点の回転、多点入力による球、球と平面の交線等17種類のルーチンから成っている。

Condor S-20
リレーショナルデータベース ¥260,000
CP/M、で稼動する本格派データベース。簡単なコマンドで専門プログラマーを必要としない。
大容量データーの高速処理可能

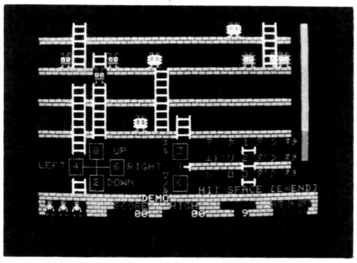
ご注文は現金書留又は銀行振込でどうぞ(銀行振込の場合は、はがきであらかじめご注文下さい)○振込先　大和銀行新宿支店普通6108825

C.C.W 株式会社 シー・シー・ダブル
〒151 東京都渋谷区代々木2-15-12-601 (03) 379−1831

技術者募集!! (SE, プログラマー)
業種別特注パッケージ開発受託

全国のFMファンにおくる

PC・MZに負けない本格的なゲームソフトを!!

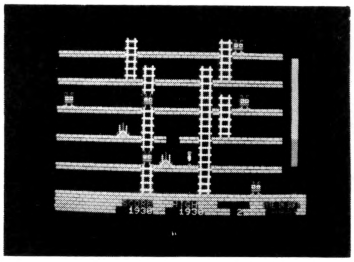


エイリアンFMパニック

FM-8・FM-7 マシン語

次から次へと出てくる エイリアンを君のスコップで退治する。

…………¥2,800

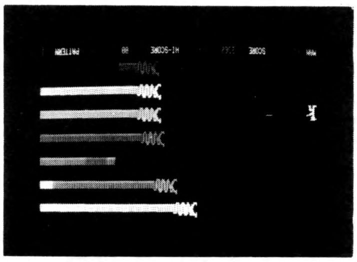


エイリアンFMクラッシュ

FM-8・FM-7 マシン語

エイリアンを 槍の上に落して退治する、第1弾のエイリアンFMパニックよりむずかしいのだ!!

…………¥2,800

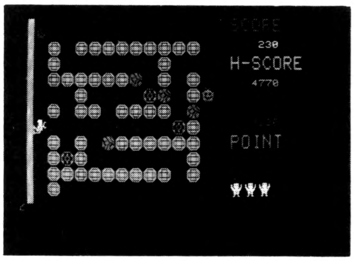


THEおろち

FM-8・FM-7 マシン語

首ののびるおろち軍団を激退せよ。

…………¥2,800



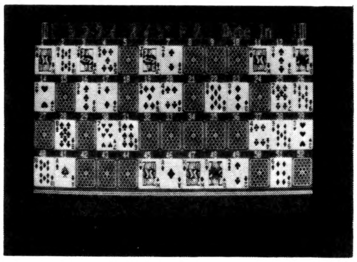
MENGO

FM-8・FM-7 マシン語

オバケを君の石けりで退治する。速いぞ!!

何面クリアできるかな。

…………¥2,800



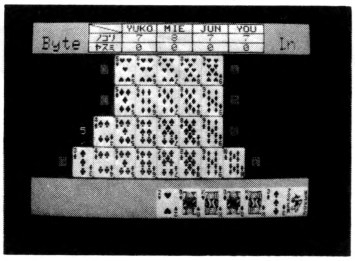
神経衰弱

FM-8・FM-7 BASIC+マシン語

君一人でする時はFMが相手になるヨ。

レベルを選んでSTARTだ。

…………¥2,800

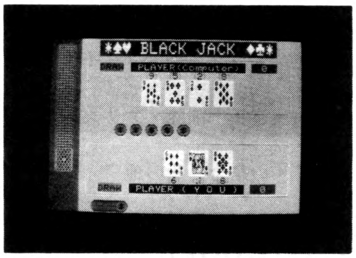


七ならべ

FM-8・FM-7 BASIC+マシン語

FMが3人分君が1人で7ならべ勝敗はジョーカの使い方。

…………¥2,800



ブラックジャック

FM-8・FM-7 BASIC+マシン語

あなたの10ドルで始まるカジノゲームでシャッフルが本格的にできている。

…………¥2,800

●FM-11用ディスク版

エイリアンFMパニック・THEおろち

エイリアンFMクラッシュ・MENGO

各…………¥3,800

●酒類販売報告書プログラム

FM-7・FM-8

MZ-80・MZ-2000…………¥4,500

MENGO MZ-80 K2E/1200 …………¥2,800

COSMO WARS …………¥2,800

ぎゃらめん …………¥2,800

販売店の方は電話にてお問い合わせください。

コンピュータショップ

Byte In

〒080 北海道帯広市西17条南3丁目(競馬場通り) ☎(0155)35-6781

★通信販売をご希望の方は、機種名、品名、住所、氏名、電話番号を明記の上、代金+送料(300円)を現金書留又は郵便為替にてお送り下さい。

トロピカルなあなたに
一流品だけをお届けする JUSCO

CVA
COMPUTER VISUAL & AUDIO

マイコン通信販売

低金利
低価格

※クレジットは1～48回までの
分割均等支払いとボーナス併用支払いがご利用できます。(初回支払いは端数がプラスされます)

NEC PC-8001mkII

①PC-8001markII……………………123,000

●20回支払(ボーナス併用)例
月々 3,300円 ㋭2万×3回
●48回均等支払例
月々 3,000円

②PC-8001markII……………………123,000
シャープ14M-111C(14インチカラーモニター)……67,800
PC-6081(データレコーダー)……12,800
カラーCRTケーブル……1,900
合計価格205,500

●20回支払(ボーナス併用)例
月々 5,600円 ㋭3万×3回
●48回均等支払例
月々 4,800円

③本体に
フロッピーディスク・プリンターをつけて

PC-8001markII……………………123,000
シャープ12M-212C(2,000文字カラーモニター)……99,800
PC-6081(データレコーダー)……12,800
東京電子LFD-550PC(ミニフロッピーディスク)……148,000
エプソンRP-80(プリンター)……89,000
PC-8037 2w(システムディスク)……7,000
プリンターケーブル……7,500
カラーCRTケーブル……1,900
フロッピーケーブル……5,000
合計価格494,000

●20回支払(ボーナス併用)例
月々 13,700円 ㋭7万×3回
●48回均等支払例
月々 11,700円

富士通FM-7

①FM-7本体……………………126,000

●28回支払(ボーナス併用)例
月々 3,400円 ㋭ 2万×3回
●48回均等支払例
月々 3,000円

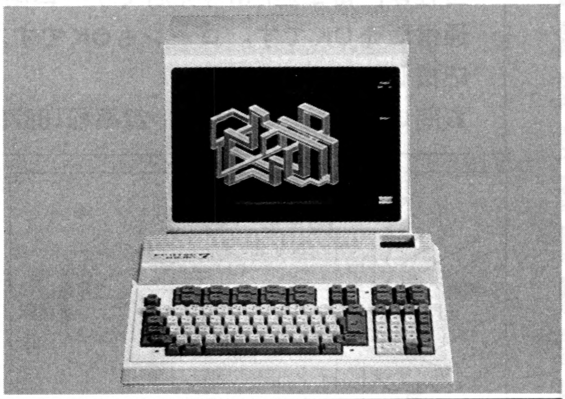
②FM-7本体……………………126,000
MB27305(12インチカラーモニター)……64,800
MR-11DR(データレコーダー)……12,800
カラーCRTケーブル……1,900
合計価格205,500

●20回支払(ボーナス併用)例
月々 5,600円 ㋭3万×3回
●48回均等支払例
月々 4,800円

③本体に
フロッピーディスク・プリンターをつけて

FM-7本体……………………126,000
MB27303(12インチカラーモニター)……99,800
MB27501(データレコーダー)……12,800
エプソンRP-80(プリンター)……89,000
MB22607(ミニフロッピーディスク)……98,000
MB22407(I/Oカード)……14,800
MB26514(プリンターケーブル)……6,800
カラーCRTケーブル……1,900
合計価格449,100

●20回支払(ボーナス併用)例
月々 12,500円 ㋭6万×3回
●48回均等支払例
月々 10,400円



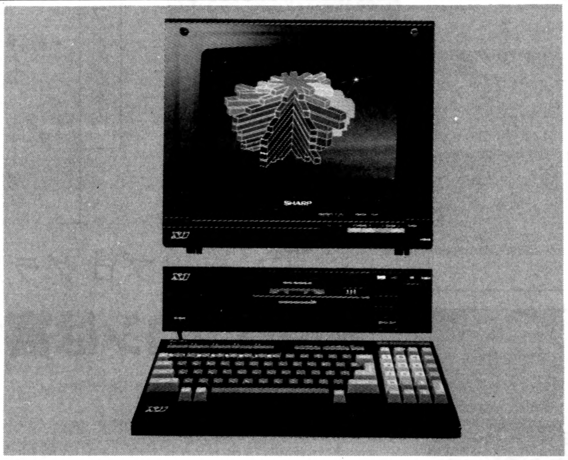
シャープ パソコンテレビ X1

①X-1本体+グラフィックRAM
CZ-800C……………………155,000
CZ-800D……………………113,000
CZ-8GR(グラフィックRAM)……32,000
合計価格300,000

●20回支払(ボーナス併用)例
月々 8,900円 ㋭4万×3回
●48回均等支払例 月々 7,200円

②本体にデジタルテロッパーをつけて
X-1本体+グラフィックRAM……300,000
CZ-8DT(デジタル・テロッパー)……89,800
合計価格389,800

●20回支払(ボーナス併用)例
月々 11,800円 ㋭5万×3回
●48回均等支払例 月々 9,300円



③本体にフロッピーディスク・プリンターをつけて
X-1本体+グラフィックRAM……300,000
CZ-800F(ミニフロッピーディスク)……198,000
CCZ-800P(プリンター)……142,800
CZ-8EP(I/Oポート)……11,800
CZ-8FA(インターフェイス)……24,000
合計価格676,600

●20回支払(ボーナス併用)例
月々 19,300円 ㋭9万×3回
●48回均等支払例 月々 15,900円

★ビジネスマンの必携!!ハンドヘルドコーナー

商品名	標準価格	20回支払例 毎月	20回支払例 ボーナス時加算額
エプソンHC-20	138,600	4,000×20回	2万×3回
NEC PC-8201	138,000	4,000×20回	2万×3回
NEC PC-2001	59,800	3,200×20回	――
カシオFP-200	69,800	3,700×20回	――
ナショナルJR-800	128,000	3,400×20回	2万×3回
東芝パソピア・mini	54,800	3,600×15回	――

大特価	商品名	標準価格	20回ボーナス併用例	均等支技例
	トミー ぴゅう太	59,800	―――	3,700×10回

電話1本で商品先どり!!

信用第一
1. 全国無料配達
2. 低金利クレジット
3. 頭金なしで商品先どり
4. 完全保証
5. 日・祭日も受付け可

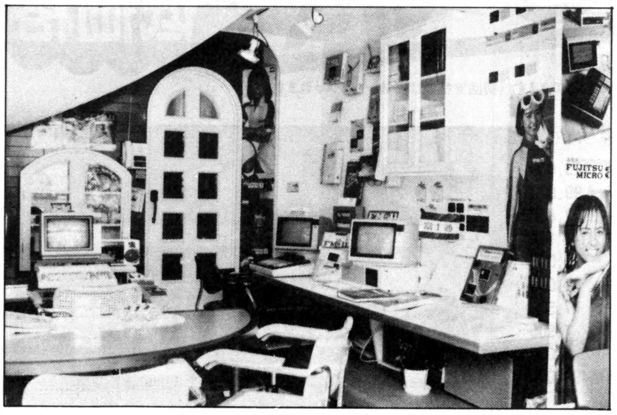
商品名	標準価格	20回ボーナス併用例	均等支払例	商品名	標準価格	20回ボーナス併用例	均等支払例
㊎東芝パソピア7	119,800	3,700×20回 ㋭1.5万×3回	3,700×36回	SONY-SMC-70	228,000	7,800×20回 ㋭3万×3回	5,900×48回
シャープMZ-731	128,000	3,400×20回 ㋭2万×3回	3,100×48回	アップルIIe	378,000	13,200×20回 ㋭4万×3回	9,200×48回
シャープMZ-721	89,800	3,200×20回 ㋭1万×3回	4,700×20回	MZ-2200	128,000	3,500×20回 ㋭2万×3回	3,100×48回
日立ベーシックマスターレベル3マーク5 台数限定	198,000	3,800×20回 ㋭1.5万×3回	3,700×36回	PC-8801	228,000	6,200×20回 ㋭3万×3回	5,100×48回
㊎日立MRAK5 MB-6892	118,000	3,800×20回 ㋭1.5万×3回	3,600×36回	ナショナル JR-200	79,800	―――	3,900×20回
シャープMZ-2000 グラフィックボード付	257,000	7,800×20回 ㋭3万×3回	5,900×48回	PC-6001 mkII	84,800	3,000×20回 ㋭1万×3回	4,500×20回

お申し込みは☎コレクトコール〈106〉番で!!
★東京(03)246-8948・8949 OH-FM係迄
〒103 東京都中央区日本橋本町1-14 丸柏ビル

ジャスコ株式会社CVA事業部

信用第1(創業大正15年) 資本金 億4,627万円
売上高 6,518億円(昭和58年2月期)

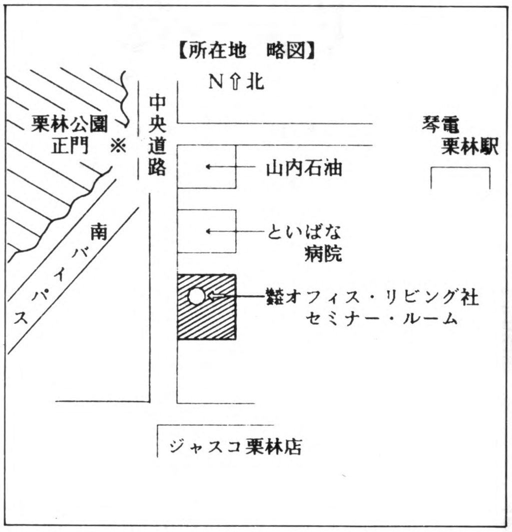
富士通FM専門店！



〔ショールーム〕

〔教 室〕

当社は，富士通FM専売店です。四国の皆さん、FMと決められたなら，ぜひ当社迄ご連絡下さい。
通信販売OKです。ローンもOKです。これからも，当社は富士通製品を愛し続けてゆきます。
応援して下さい!!
お問い合わせ、ご相談は、☎高松(0878)61-7111お気軽にどうぞ。



FMのことなら

(株)オフィス・
リビング社

四国・高松

FMユーザー必携品　FM-7,8用

オリジナル・ソフトのご案内

- 絵浮恵夢　日本語ワープロ　¥4,500
  日本語のメッセージに従って操作出来ます。
  カセットの限界に挑戦した日本語ワープロ。
- 絵浮恵夢　メモリチェンジャー　¥3,500
  マシン語入力時間の大幅短縮を！！
  三つの入力モードを選べます。縦横のチェックサムを見ながらカーソルエデットが。
- 絵浮恵夢　タイピング　¥3,500
  柔らかい電子音にのって，タイプとF-BASICの勉強を！！　ピッチ音は自在に設定。
- 絵浮恵夢　変数リスト　¥3,500
  オールマシン語の高速処理です。ABC順で変数の使用行№と回数を表示します。プログラムを組みだすと，必ずや必要に。
  (11用もあり)

今，FM専用プログラム募集中！

今やパソコンは富士通の時代です!!

数ヶ月前に発売されて以来，驚異的に売れているパソコンFM-7はじめ，充実のFM-8，ビジネスマン・企業家の為のFM-11等これからのパーソナルコンピューターの世界をリードしていく３機種が注目の的です。

当社は，ショールーム・教室・FMユーザーの同好会を運営する四国で唯一の**富士通FM専売店**です。各機種の展示をはじめ，周辺機器・書籍・プログラム他すべてが揃います。又，ご希望でレンタル貸しも実施しております。

もし，富士通のパソコンと決められたなら，当社の様な専売店でご購入されることが，安心して使いこなしてゆく又，最新の情報を得る決め手となります。ぜひ一度当社にいらっしゃいませんか。**電話でのお申込も受付中**です。

富士通パーソナルコンピューター専売店
株式会社 オフィスリビング社
〒760 高松市栗林町一丁目16番21号　TEL 0878(61)7111㈹

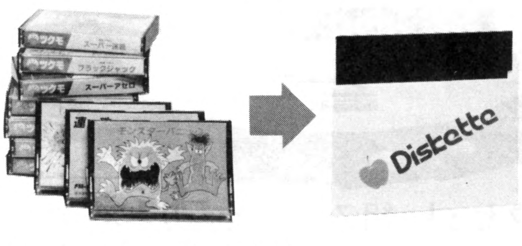
大好評ツクモオリジナルユーティリティー…買ってて良かった

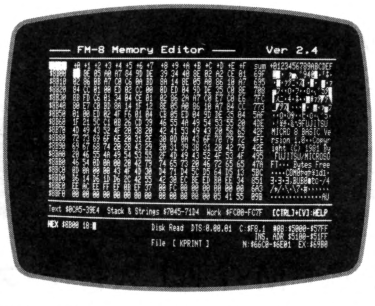
FM-7/8 なんでもかんでもユーティリティー

従来のなんでもかんでもユーティリティーがFM-7でも使えるようになりました。

FM-7/8共用ディスケット1枚 取扱い説明書付 ¥9,800送料¥500

1枚に10種類入ります。





1 ゲームローダーとして
今までのDISKに入らなかったテープ版のゲームソフトが1枚のディスクに10種類入ります。

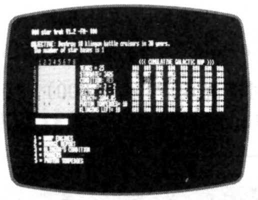
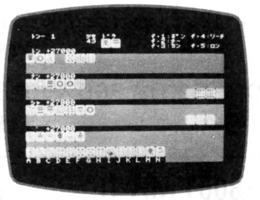
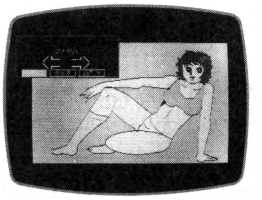
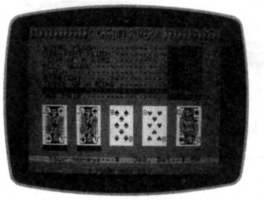
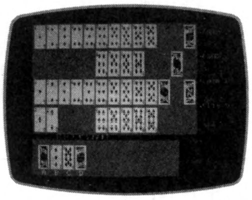
2 メモリー、DISKエディターとして

※その他の機能 ○はFM-8のみ

◎指定トラックコピープログラム
◎連続フォーマットプログラム
◎アスキー表示付・高速ダンプ
◎アスキー表示付・ディスクダンプ
◎プリンタのイニシャライズ
80文字／132文字、66行／88行ページ指定付
◎文字列、及びバイナリーコードのサーチ、及び書き替え
◎マシン語プログラムのアドレス表示
◎ディスクファイルなんでも転送
◎ゲームローダーで作ったディスクのオートスタート
◎BASICのプロテクト解除
○フリーエリアが2K広いBASIC
○オートスタートディスク作成
○オールRAMモード用システムディスク作成

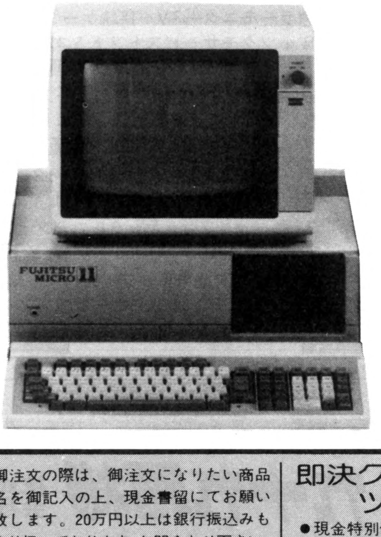
大好評 ツクモオリジナルFM-11ソフト5本パック

DISK版 特価¥9,800

スーパースタートレック　スーパー4人麻雀　野球拳　カジノポーカー　七並べ

本格ビジネス用 オリジナル組み合わせシリーズ



富士通 FM-11組み合わせシリーズ

FM-11AD＋漢字ROMカード＋14インチ高解像カラーモニター（640×400ドット）＋ケーブル＋10インチ・シリアル・ドット・プリンタ＋ケーブル
＝定価￥617,300➡特価￥**524,000**
FM-11ADをEXタイプに変更の場合＝定価￥677,300➡特価￥**570,000**

- FM-11には、簡易言語「FM CALC」が付属しており、本格的ビジネス分野に利用できます。
- 漢字ROMカードには、日本語ワード・プロセッサーが付属しています。
- ツクモ電機でFM-11をお買い上げのお客様には、DISK版ゲーム（5種類 スーパー4人麻雀etc）￥9,800をプレゼント致しております。

御注文の際は、御注文になりたい商品名を御記入の上、現金書留にてお願い致します。20万円以上は銀行振込みも取り扱っております。お問合わせ下さい。

富士銀行　神田支店
普通口座　№894047

即決クレジット
ツクモ全国クレジットOK!!

- 現金特別価格で各種クレジットが利用できます。残金のみに金利がかかります。
- 60回払いまで出来ます。但、1回の支払い額は3,000円以上。
- その場で、お持ち帰りできるクレジットもあります。
- 印鑑、身分証明書（免許証など）が必要です。尚、学生の方及び、未成年の方は、ご両親の保証も必要です。
- 取扱いクレジットカード：日本信販、JCB、DC、UC、VISA、MC、CF

通信販売ご利用の方は、電話番号、年令も必ずご記入下さい。

■通信販売は　〒101 東京都千代田区神田郵便局私書箱135 九十九電機㈱　FM係

九十九（ツクモ）電機 株式会社

ニューセンター店	〒101	東京都千代田区外神田1-16-10	☎03(251)0986～8
名古屋店3F	〒460	名古屋市中区大須3-30-86 ラジオセンター・アメ横ビル内	☎052(263)1681
5号店	〒101	東京都千代田区外神田3-1-14	☎03(251)0531～2
7号店	〒101	東京都千代田区外神田1-9-9	☎03(253)4199
札幌店	〒060	札幌市中央区南二条西3-15-1 さっしんビルB1	☎011(241)2299

■定休日　東京各店は毎週木曜日と第3水曜日、名古屋店は毎週月曜日

セットで買えば、衝撃的に安くなる。

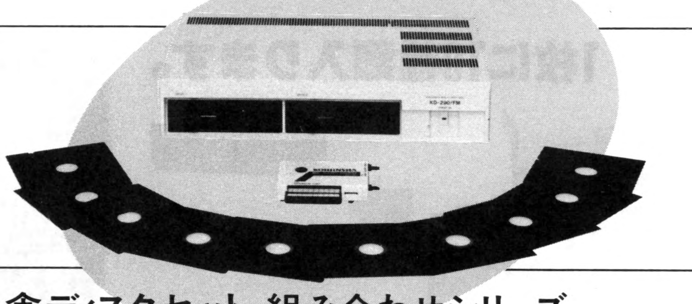
ツクモは買い物天国だ。じっくり選べばこの夏の旅費ぐらい軽く浮いちゃうかも……
ローンもOK、狙い目はツクモで!!

FM-7用 工人舎ディスクセット

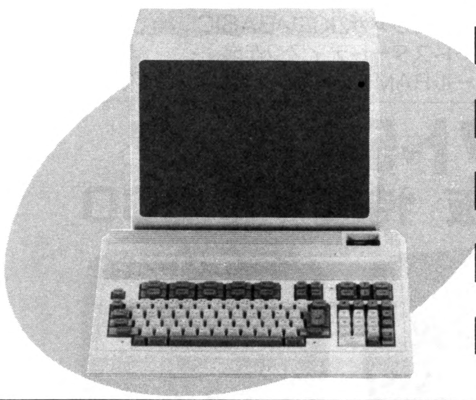
KD290/FM＋コントローラー＋システムディスケット
＋生ディスケット10枚…

定価 ￥190,000⇒特価 ￥152,000

※FM-8も同金額にて組み合わせが可能です。注文の時には7・8の別を御指定下さい。



富士通FM-7＋工人舎ディスクセット 組み合わせシリーズ



Aセット FM-7＋工人舎ディスクセット＋家庭TV用アダプター＋DISK版オリジナルソフト5本パック
定価￥337,800⇒特価 ￥275,000

Bセット FM-7＋工人舎ディスクセット＋中解像カラーモニターTV＋接続ケーブル＋DISK版オリジナルソフト5本パック
定価￥392,400⇒特価 ￥305,000

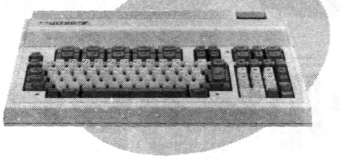
Cセット FM-7＋工人舎ディスクセット＋NH14DD（中解像2000文字）＋接続ケーブル＋オリジナルソフト5本パック
定価￥397,400⇒特価 ￥317,000

Dセット FM-7＋工人舎ディスクセット＋NH12D9（高解像2000文字）＋オリジナルソフト5本パック
定価￥425,600⇒特価 ￥339,000

Eセット FM-7＋工人舎ディスクセット＋C14-452（日立RGB入力端子付家庭用TV￥79,800）＋接続ケーブル＋オリジナルソフト5本パック
定価￥407,400⇒特価 ￥317,000

富士通FM-7 組み合わせシリーズ

￥126,000（簡易言語ソフト付）



Aセット FM-7＋家庭TV用アダプター＋専用データレコーダー＋ツクモサービスソフト5本
定価￥165,800⇒特価 ￥133,000

Bセット FM-7＋中解像度カラーモニターTV＋接続ケーブル＋専用データレコーダー＋ツクモサービスソフト5本
定価￥220,400⇒特価 ￥163,000

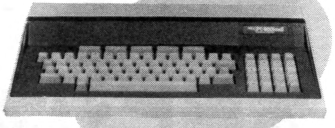
Cセット FM-7＋NH14DD（中解像2000文字）＋接続ケーブル＋専用データレコーダー＋ツクモサービスソフト5本
定価￥225,400⇒特価 ￥175,000

Dセット FM-7＋NH12D9（高解像2000文字）＋専用データレコーダー＋ツクモサービスソフト5本
定価￥253,600⇒特価 ￥197,000

Eセット FM-7＋C14-452＋接続ケーブル＋専用データレコーダー＋ツクモサービスソフト5本
定価￥235,400⇒特価 ￥175,000

NEC PC-8001 MK II 組み合わせシリーズ

￥123,000



Aセット PC-8001 MKII＋家庭TV用アダプター＋データーレコーダーサンヨー11DR＋ツクモサービスソフト5本
定価￥164,300⇒特価 ￥131,000

Bセット PC-8001 MKII＋中解像カラーモニターTV＋接続ケーブル＋データレコーダーサンヨー11DR＋ツクモサービストフト5本
定価￥217,400⇒特価 ￥163,000

Cセット PC-8001 MKII＋NH14DD（中解像2000文字）＋接続ケーブル＋データレコーダーサンヨー11DR＋ツクモサービスソフト5本
定価￥222,400⇒特価 ￥175,000

Dセット PC-8001 MKII＋NH12D9（高解像2000文字）＋データレコーダーサンヨー11DR＋ツクモサービスソフト5本
定価￥250,600⇒特価 ￥197,000

Eセット PC-8001 MKII＋C14-452＋接続ケーブル＋データレコーダーサンヨーMD11R＋ツクモサービスソフト5本
定価￥232,400⇒特価 ￥175,000

Fセット PC-8001 MKII＋PC-8053（14インチ高解像￥198,000）＋接続ケーブル＋データレコーダーサンヨー11DR＋ツクモサービスソフト5本
定価￥350,600⇒特価 ￥209,000

＊データレコーダーをKD290に変更ご希望の方は￥107,000を加算した額が各合計特価になります。

御注文の際は、御注文になりたい商品名を御記入の上、現金書留にてお願い致します。20万円以上は銀行振込みも取り扱っております。お問合わせ下さい。

富士銀行 神田支店
普通口座 №894047

即決クレジット ツクモ全国クレジットOK!!

- 現金特別価格で各種クレジットが利用できます。残金のみに金利がかかります。
- 60回払いまで出来ます。但、1回の支払い額は3,000円以上。
- その場で、お持ち帰りできるクレジットもあります。
- 印鑑、身分証明書（免許証など）が必要です。尚、学生の方及び、未成年の方は、ご両親の保証も必要です。
- 取扱いクレジットカード：日本信販、JCB、DC、UC、VISA、MC、CF

通信販売ご利用の方は、電話番号、年令も必ずご記入下さい。

■通信販売は 〒101 東京都千代田区神田郵便局私書箱135 九十九電気(株)KFM8係

九十九電機 株式会社

ニューセンター店	〒101	東京都千代田区外神田1-16-10	☎03(251)0986〜8
名古屋店3F	〒460	名古屋市中区大須3-30-86 ラジオセンター・アメ横ビル内	☎052(263)1681
5号店	〒101	東京都千代田区外神田3-1-14	☎03(251)0531〜2
7号店	〒101	東京都千代田区外神田1-9-9	☎03(253)4199
札幌店	〒060	札幌市中央区南二条西3-15-1 さっしんビルB1	☎011(241)2299

■定休日 東京各店は毎週木曜日と第3水曜日、名古屋店は毎週月曜日

新発売

FM-8&FM-7 Disk Utility

HOSPITAL

健康診断
ディスケット上に機械的なキズ(Physical Error)がないかどうか調べます。全トラックについて読みだしテストを行ない、必要に応じてFormat・システムコピー等をする事が出来ます。

胃腸科
F-Basic用ディスクのFAT/Direetryを解読します。FAT/Direetryの指定のほかに"FILES"の指定をすると、Killされたファイルも含めて、ファイル名・クラスタ使用状況の一覧表が示されます。

X線診断
FM-8仕様に限らず、両倍(40T)・片倍(35T)ディスケットの任意のセクタがスクリーン上に示されます。他社仕様のディスクを読んで理解するためには、その社の呼称法、セクターの読み出し順序の法則を、F-Basicでの法則に換算しながら進めて下さい。

整形外科
上と同じく、任意のセクタをバイト単位でスクリーン上で修正できます。このルーチンも無意味なトラックを指定しない限り他社仕様のディスクも対象となります。間違って入力しても(CR)しない限り何度でも修正出来ます。

脳神経外科
プログラムの収納の構造が調べられ、ツブれたプログラムの修復・他社仕様のディスケットの移植の準備に使われます。各クラスタの最初のセクタを表示します。また、このセクタの前半分と直前のクラスタの最後の半セクタを表示することや、直前のセクタのつながりを見たり、プリンタがあればCopyを取ったり、新しいクラスタの指示も行なう事が出来ます。

価格 15,000円 〒300

産婦人科
F-Basic用のFormat・システムコピー・初期化とVolcopy等を実行します。Format・Volcopyは、他機種、他言語にも使えます。いずれの場合も、#0上のプログラムディスケットが読み込まれ実行が始まります。#0に入っているプログラムディスケットを、出来るだけ出し入れしないようにしていますのでFujitsuから提供されるものよりは作業が簡略化されています。

人工受精
片倍(35T)ディスケットの任意の面の全Copyに使います。表面(例えばPC仕様の片面)、裏面(例えばIf-800)が指定できます。片倍(35T)のディスケットは普通反対面が、Format していないので、通常のVolcopyが出来ません。この様な場合、表あるいは裏を指定して、その面を35トラックだけVolcopyするルーチンです。尚、PASOPIAのように両面で35Tの場合は、このルーチンを表裏2度走らせば良いのです。このプログラムでも、産婦人科でもVolcopyは、Killされたファイルも含めて、内容を判断する事なく、完全Copyします。

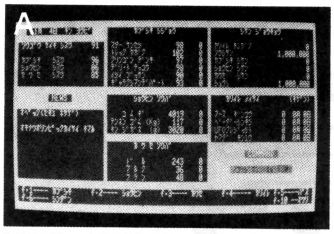
心臓移植
PC-8801(NEC)の両倍仕様の、任意の一つのプログラム・ファイルをFM-8仕様にCopyします。それに先立ってPC-8801仕様のディスケットの構造を、調べる事が出来ます。このルーチンのうちファイルコピープログラムそのものは、オリジナルがAscii Saveされていなくても動きますが、目的が移植にあるわけですから、オリジナルのAscii Saveは必項です。このプログラムではドライブ番号、トラック番号等をいきなりFM-形で書かないで、一度PC(両面)形で書いて改めて換算するようにしています。また、片面倍密度仕様のディスケットでも使用出来ます。

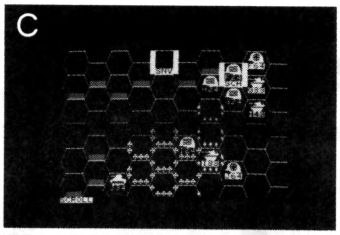
健康手帳『VIVA FM ディスク』自費出版発売中!(3,000円)

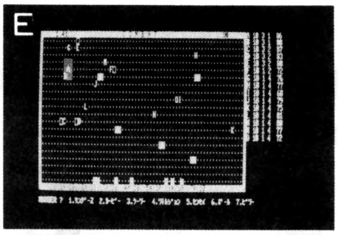
健康手帳『ディスクインウィークス』を改題し『VIVA FM ディスク』としました。
健康手帳『ディスクインウィークス』をお持ちの方追加発表No.2ができました。送料共¥1,200

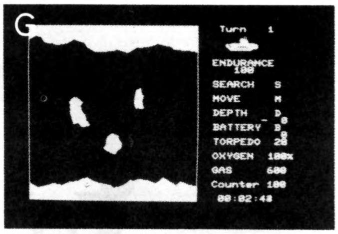
テープ・ロード・アダプター
テレコとマイコンの間に接続して、波形を成型するので、LOADミスがほとんど、生じなくなります。
価格 8,800円(〒200)

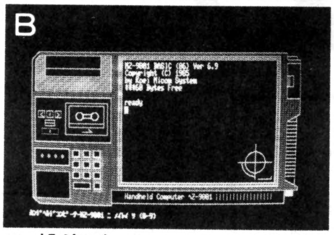
WORLD WIDE COMPUTER SUPER SHOP
COSMOS™ 岡山
〒700 岡山市南方5-6-5 今田ビル2F(県営グランド入口向)
TEL(0862)54-7474　ミニファクス(0862)54-7481
年中無休 AM10:00〜PM7:00
通販は現金書留又は郵便振替 岡山4-12524 コスモス岡山

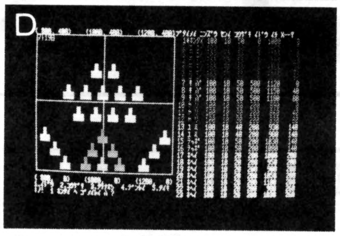


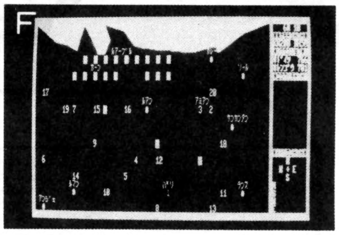


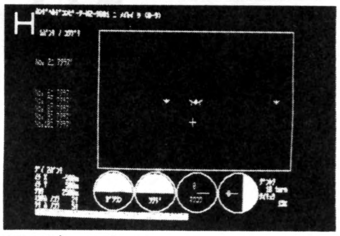












A. 投資ゲーム　　¥5,000

シミュレーションゲーム

ソ連大凶作、金放出か、中東に戦火、ウォール街の動きに東京市場は……？貴方の分析はいかに。
●機種／PC-8001、PC-8801、FM-7・8、MZ-80B(HuG-BASICとグラフィックラム1必要)、レベルIII。●種類／テープ

B. 地底探検　　¥4,500

シミュレーションゲーム

たび重なる危機、おそいかかる怪獣、地底人をなぎ倒し財宝を手に入れる事が出来るか……
●機種／PC-8001、PC-8801、FM-7・8、レベルIII。●種類／テープ

C. バルジ大作戦　　¥4,500

シミュレーションウォーゲーム

ドイツ軍が命運を賭けた最後の決戦。ヘスク画面が前後左右にスクロール。スゴイ!!
●機種／MZ-80B。●種類／テープ

D. 川中島の合戦　　¥3,500

シミュレーションウォーゲーム

貴方は武田信玄。騎馬・弓・歩兵他の部隊を総指揮して、大挙して押し寄せる上杉勢に勝てるか
●機種／PC-8001、PC-8801、FM-7・8、MZ-80B(HuG-BASICとグラフィックラム1必要)、レベルIII
●種類／テープ

E. コンバット　　¥4,500

シミュレーションウォーゲーム

貴方はサンダース軍曹。カービー、ケリー、リトルジョン達を率いて、独軍陣どる丘の上のトーチカを攻略
●機種／PC-8001、PC-8801、FM-7・8、レベルIII、MZ-2000 MZ-80B（グラフィックラム1必要）
●種類／テープ

F. ノルマンディー上陸作戦

シミュレーションゲーム　　¥4,500

1944年Dデイ ノルマンディーより上陸し、ベルリンに向い大進撃を開始した連合軍を貴方(独軍参謀長)は食い止められるか
●機種／PC-8001、PC-8801、FM-7・8、MZ-80B、MZ-2000（グラフィックラム1必要）●種類／テープ★ボード別売……¥1,200＋送料(¥300)

G. ダス・ブート　　¥3,500

シミュレーションウォーゲーム

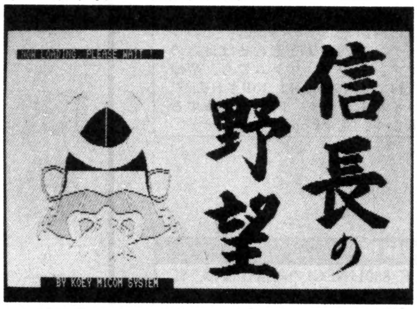
Uボート艦長である貴方の使命は輸送船団に壊滅的打撃を与える事だ　敵護衛艦との攻防はいかに！(グラフィック機能豊富)
●機種／PC-8001、PC-8801、FM-7・8、MZ-80B(HuG-BASICとグラフィックラム1必要)、レベルIII　●種類／テープ

H. 空中戦　　¥3,500

シミュレーションウォーゲーム

各種の飛行機をあなたが選び戦闘を展開
●機種／FM-7・8、PC-8801　PC-8001
●種類／テープ

ゲームデザイナー：シブサワコウ

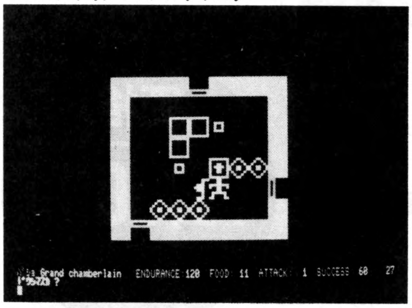


■信長の野望　　¥4,500

ウォーゲームシリーズ

全国各地を猛将が割拠する戦国時代慧星の如く出現し、天下統一という野望を胸に戦さを挑む、戦国の革命児、織田信長彼の人生は戦さの連続だ！今、貴方は「信長の野望」を果たせるか！●機種／PC-8801。●種類／テープ
PC-8001、FM-7・8。

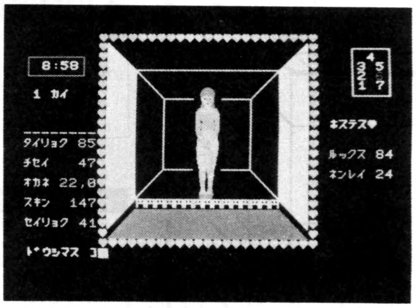
ゲームデザイナー：シブサワコウ



■クフ王の秘密　　¥4,500

ロールプレイングゲーム

エジプト最大を誇るクフ王のピラミッド　ここには莫大な財宝が隠されている。もしこの財宝を探し当てることが出来たら、たちまち大富豪だ！だが、今貴方の手元には手掛かりとなるものは何もない。さぁ、この財宝さがしに貴方はどう挑むか！●機種／PC-8001。●種類／テープ



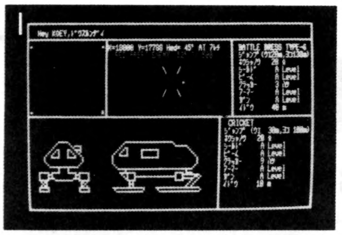
■団地妻の誘惑　　¥4,800

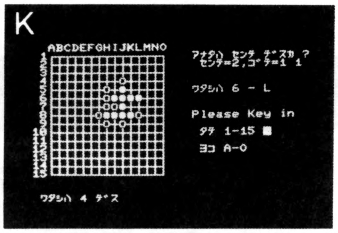
アダルトポルノロールプレイングゲーム

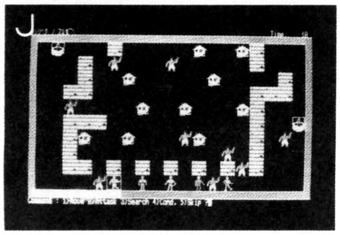
貴方は避妊具のセールスマン、ニュー越谷団地に狙いを定め戸別販売を開始した。団地妻の誘惑、女子大生の色気、ヤクザのおどし等を乗り越え避妊具を売り歩く貴方。時間はどんどん過ぎてゆくし、体力も消耗してきた。さぁ、数々の誘惑に打ち勝ち全ての避妊具を売り切れるか！
●機種／PC-8801。●種類／テープ

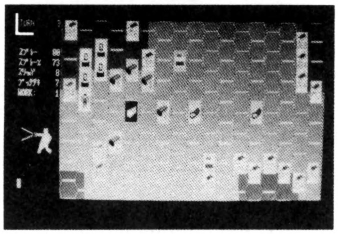
バラエティーソフト
NEW パッケージ

VARIETY SOFT
創造領域に限界なし。さらに磨きをかける光









I. サンセット・イン・ラディック
SFシミュレーション　¥4,500
敵の基地に忍びこみ、機密文書を盗み出せ、これが貴方にあたえられた指令だ　G型バトル・ドレスを身にまとい、今戦いが始まる………
●機種／PC-8001、PC-8801、FM-7・8、LIII
●種類／テープ

J. ドラゴン&プリンセス ¥4,500
ファンタジーアドベンチャー
王国より選び抜かれし戦士よ、今こそ城の宝を取りもどせ、報酬は姫と金貨にあり　さぁ、冒険ファンタジーの世界で君の技をためす時だ
●機種／PC-8001、PC-8801、レベルIII、FM-7・8　●種類／テープ

K. 連珠　¥3,500
日本古来の知的ゲーム
別名五目ならべとして親しまれ、15目の盤上に戦われる高度なテクニック　さぁ、貴方はこの勝負にどう挑むか…
●機種／PC-8001　●種類／テープ

遂に完成!
L. ホイホイ　¥3,500
パロディ・シミュレーションゲーム
部屋の中に20ヒキの、あのニックきゴキブリドモか我がもの顔にウロチョロ…　君！だまって見てないで、部屋にとびこんで片っぱしからヤッつけろ！君の武器はスリッパとスプレー、それにホイホイと実に多彩　ゴキブリドモは例によってかなり挑発的だ　ゴキブリの頭が勝つか、君の頭脳が勝つか、さぁどっち（これはあくまでも冗談のゲームですので念のため）
●機種／PC-8801　●種類／テープ

PC-8001で 漢 字 いろは

● PC-8001用
日本語 ワードプロセッサー
●基本プログラム……¥16,800
●熟語辞書(オプション)‥¥9,800
●縦書き機能(オプション)¥4,500

●マシン語による高速処理●漢字ROM不要●漢字プリンター不要●漢字タブレット板不要●本体改造不要。

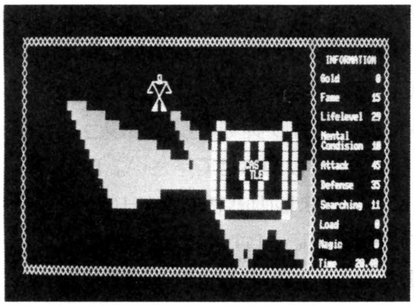
特徴
▶仮名漢字変換方式▶熟語変換可能(辞書部オプション)▶拡大文字、アンダーラインプリント可能▶カーソルエディット方式による豊富なエディター（挿入、削除訂正、右よせ、センターリング他）▶JIS第一水準漢字を内蔵▶片仮名、平仮名、英数記号、ギリシャ文字、ロシア文字その他を内蔵▶外字登録可能▶美しいプリンター出力（16×16ドット構成で普通文字文）▶縦書き出力可能(オプション)。

使用可能プリンター
PC-8023、PC-8821、PC-8822、MP-80II、MP-80III、RP-80、GP-250F。

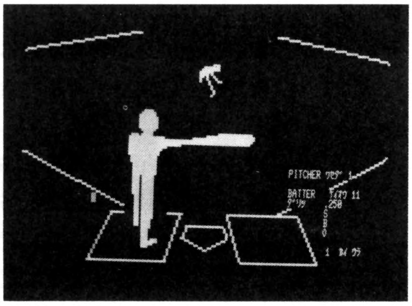
お求めは全国マイコンショップ又は本社あてに現金書留(送料サービス)に機種を書いてお送り下さい。

絶賛発売中!
FM7・8用 ビジネスソフト

汎用漢字顧客管理システム
■商売繁盛　¥19,800
顧客の情報が必要な時に誰れにでも簡単に得られます。最大600件の顧客情報(名前、住所、電話、年令、性別等)が仮名漢字変換方式により漢字で登録。条件に適合した顧客を自動的にセレクトも　さらにはDMの漢字宛名書きもらくらくこなします。商売繁盛には「商売繁盛」が最有力のスタッフ！
●機種／FM-7・8●種類／ディスク。



■剣と魔法　¥4,500
ファンタジー・ロールプレイング・ゲーム
冒険を求めて旅に出たあなたは国境で、邪悪な魔法使いが王を殺し、王位を奪い、しかも王女を幽閉している、という話を耳にします　血気盛んなあなたがこの話を聞き捨てるわけがありません。魔法使いを倒し、王女を救い出そうと決心し、魔法使いに敢然と戦いを挑みます。しかしその前途には思わぬ波乱と、障害の数々が…。
●機種／PC-8001。●種類／テープ



■ペナントレース　¥4,500
スポーツシミュレーションゲーム
プロ野球12球団、主力全選手のデータから、二人で互いに好きなチームを選択し、先発メンバーを編成します。メンバーが決ったらプレイボール。アニメ化された画面の中でゲームはエキサイティングに展開。もちろん作戦を選手に指令するのは監督であるあなたご自身。選手の交替もルールにそって自由にできます。小さな作戦ミスが命とりになる場合も…。冷静な判断で相手をやっつけて下さい。●機種／PC-8001。●種類／テープ

光栄マイコンシステムは、国内で初めてシミュレーションゲームを開発、シリーズ化いたしました。

栄ソフトウェア群。

■ナイトライフ
¥4,500
アプリケーション
これほど実用的で家庭に密着したソフトはない。その日の体調にあったフォルム、同時に安全日もお教え致します。豊かなナイトライフをお過し下さい。（結婚祝いにもどうぞ）●機種／PC-8001　●種類／テープ
（ディスクは開発中です。）　PC-8801、FM-7・8。

ゲームパックシリーズ
PC-8001、PC-8801用各¥3,800
（PC-8801用はSilver Pack IIのみ）

●**Silver Pack II**
▶モグラたたき▶スターキャッチャー▶スタートレック▶テニス▶スネーククネクネ▶イーブン▶フィーガーショック▶さいころ賭博▶クラッシュ▶ACYデューシー▶リバース。

●**Golden Pack II**
▶ミク25▶藤原京エイリアン▶ラビットハンター▶15ゲーム▶ザ・ブリズナ▶センチピード▶2次元迷路▶ハノイの塔▶ルーレット▶恋うらない▶ビッド&ブロー。

●**Bronze Pack 6**
▶アレンジボール▶国盗ゲーム▶三次元迷路▶ダサク▶バドルオブタンク▶スペースチェス。

●**Platina Pack 5**
▶チェッカー▶ディグ・ダウン▶ニム▶山くずし▶ゴルフ。

光栄マイコンシステム★
本　社：〒326 足利市通1丁目2677　☎0284-41-5911代　日・祭日休
足利店：〒326 足利市通1丁目2677　☎0284-44-1581（11:00～20:00）年中無休
日吉店：〒223 横浜市港北区日吉本町1876　☎044-61-6861（10:00～19:00）日・祭日休

EDITOR'S ROOM

◆FMのハード・ソフトに詳しい人で，原稿の書ける方を募集しています。簡単な自己紹介，マイコン歴，得意な分野，実力評価用の作品（ハードまたはソフト）など，原稿依頼の資料となるものを郵送するか，電話のうえ直接ご持参ください。

〒102 千代田区四番町2-1 ㈱日本ソフトバンク
Oh! FM編集室 松岡 ☎03(261)4095

◆オリジナルプログラムも下記の要領で募集しています。

- ●①プログラムの内容，特徴，アルゴリズムなどの説明，②フローチャート・変数表，③遊び方や操作方法，注意事項 ④プログラムの入ったテープないしはディスケット（機械語の場合は，ソースプログラムも）以上と，住所・氏名・年齢・職業・電話番号などを明記のうえ，郵送するか直接お持ちください。あて先は上記と同様です。
- ●プログラムは未発表のものに限ります。
- ●応募原稿は原則としてお返し致しません。
- ●掲載にあたって多少修正させていただくことがあります。
- ●掲載の場合は，本誌規定の原稿料をお支払い致します。

◆本誌の内容に関して，電話による質問を受け付けます。

☎03(265)5789 受付時間 16:00〜18:00

質問はバグ情報についてのみ受け付けます。機械の取り扱いなどについてのお問い合わせはご遠慮ください。マニュアルを熟読してください。

バックナンバーのお知らせ

Vol. 1 品切れ

Vol. 2 本格派パソコンFM-11の全容／各社フロッピディスクユニットの比較検討／高速3Dグラフィックス／タイプ練習プログラム／6809マシン語入門(後)／FM-7回路図と解説(前)／ゲームリスト(さがしものゲーム，川下り，ウエディングレース) 他

Vol. 3 マシン語特集（マシン語って何んなのですか?，Oh! FM用チェックサム，中間色PAINTもどきプログラム，サブシステムプログラム転送・実行など）／三次元パッケージのアルゴリズム／初心者のためのF-BASICプログラミングアプローチ／FM-8の命令を増やそう／ゲームリスト(B-1，アルデバラン) 他

●バックナンバーご入用の際はお近くの書店にお申し込みください。
●次号は9月18日発売予定です。

●**編集** 渡辺妙子 松岡真理
●**技術** 松田辰夫 小林初雄 徳永 聡
●**海外協力** HONG・LIANG・LU IAN・ALLEN RONALD・BILLINGS HISAYOSHI・MIKAMI
●**協力スタッフ** 木下淳博 桑原岳夫 鶴岡哲明 長沼孝仁 西村義孝 本郷誠司 林 剛正
●**写真** 浜崎 昭 杉山和美
●**レイアウト** 創美レイアウト レモンデザインルーム㈱

●広告索引●

アーバン電子	60	数研塾	97
アイテム	134	ソフトマート	133
ADOショップ	121	チャンピオンソフト	122〜125
アンプルソフトウェア	カラー4	九十九電機	139・140
エイセル	カラー12	日本情報システム	116
エイト電気	128	バイトインマイコンショップ	136
エスピーシー	126・127	ハドソン	カラー8
エプソン	カラー5	ぱそる	カラー6
大阪CSK	カラー9	ピーシーエー	131
オフィスリビング	138	富士通	表2・4，カラー1・2・3
関東電子	カラー10・11	富士通興業	表3
光栄マイコン	142・143	マイクロキャビン四日市	129
コスモス岡山	141	マイコンセンター60	130
コンピュータ11	132	理工産業	120
シー・シー・ダブル	135	YDKシステムセンター	カラー7
ジャスコ	137		

（五十音順）

編集後記

■今回の表紙いかがでしたでしょうか。西海岸の夕べ，渚にたたずむ2人をイメージして描いたものです。マシンはFM-11です。8月末までに表紙サービス係へ，2000円分の切手を添えてお送りくだされば，ディスケットに入れてサービスいたします。（N）
■表紙描いた恋愛経験なしのN最近メタメタにフラれた根暗だ☺
■夏休みに入り，時間が余っている人，逆に急がしくなった人も多いと思う。FMのおもしろい活用，アイデアを，フレッシュな頭脳の皆さんに期待する。編集室にも，ぜひ知らせて下さい。（K）
■マンガ雑誌などに「○○先生に，はげましのおたよりを出そう」というのがありますね。で，わが編集長にもはげましのおたよりをください。もう編集後記は書かない，なんてなぜかすねてしまったのです。どうか温かい愛の手をさしのべてください。（M）
■大道芸というのがありますが，とてもよい!! 特に舞踏。舞っている人の表情がいいんですョ。みんなとてもいい顔してる。ああいう顔は凡人にはありません。観ている方がバカみたいに思えてくるから不思議です。どっちが人間らしいんでしょうか。(T)

隔月刊 **Oh!FM** 第4号
●昭和58年8月1日 発行
●発行人／孫 正義
●編集人／田鎖洋治郎
●発行元／株式会社日本ソフトバンク
本社／〒102 東京都千代田区九段南2-3-14 靖国九段南ビル2F
☎03(263)3690 Fax.03(263)3660
出版部／〒102 東京都千代田区四番町2-1 ☎03(261)4095
大阪支店／〒542 大阪市南区難波千日前5-19 河原センタービル
☎06(644)0191 Fax.06(644)0160
●印刷 図書印刷㈱ ☎03(453)2550

©1983 JAPAN SOFT BANK 雑誌02199-08
★本誌からの無断転載を禁じます。

富士通興業の充実したビジネスソフトウェアが FMシリーズを一段と使いやすくします。

究極のパソコンワープロ

新バージョン FM-WORP ver 2.0 定価：¥60,000

発売以来使い易さで好評の「FM-WORP」がバージョンアップされました。パソコンでは初めての文節単位の文章入力等一段と使い易さにみがきがかかりました。「FM-WORP」
「FM-WORP」は親切設計、見て、さわってお選び下さい。

- 文節単位の文章入力、しかも辞書の学習機能付
- B4/A自由自在、あなたのプリンタに合せて任意の書式が設定できます。しかも自動ケイ線機能付
- 文章名など、会話は全て日本語、入力項目は標準値採用など初心者まで楽々操作できます。

《旧バージョンをお持ちのお客様へ》
「FM-WORP」の旧バージョンをお持ちのお客様で新バージョンへのレベルアップを希望される方は旧バージョン一式に手数料5,000円を添えて当社までご送付下さい。折返し新バージョンを返送させていただきます。

新発売 三次元グラフィックス PERS-F2 定価：¥80,000

パソコンパースは、現実に存在しないイメージや、現実にあっても撮影不可能な世界、またエスキース段階のプランを映像化したり描写するものです。
本プログラムパッケージ〝PERS—F2〟は、これらの処理に必要な機能を解析し、簡便さ、精度、イメージの拡大、を目的に設計されています。

富士通興業マイコンスクール
FM-LABO
FM・LABORATORY

FMシリーズを最も良く知っているインストラクターが指導するFM-LABO。詳しい案内パンフレットができておりますので、下記のパソコン営業部03-567-3468までどうぞ。

大阪 OA PLAZA 開催中

大阪地区パーソナル・コンピュータスクール開催中です。
会場：大阪市北区堂島1-5-17 堂島グランドビル5F
富士通興業株式会社 OA PLAZA TEL06-343-2626
日程：コースなど詳細については、電話でお問い合わせください。

躍進する富士通グループ
富士通興業

富士通興業株式会社

FMシリーズのお問い合わせは……

OA機器営業本部
〒106 東京都港区六本木4-1-4黒崎ビル ☎586-1511

パソコン営業部
〒104 東京都中央区銀座2-6-1中央銀座ビル☎567-3468

札幌営業所	☎011-221-8501	大阪電子営業所	☎06-343-2626
仙台電子営業所	☎0222-62-5252	広島営業所	☎0822-22-6141
北関東電子営業所	☎0486-41-1747	九州電子営業所	☎092-472-4111
名古屋営業所	☎052-211-5866	熊本営業所	☎0963-55-3166
横浜営業所	☎045-311-4541	神戸営業所	☎078-392-1481

Oh!FM Vol.4
昭和五十八年八月一日発行　第一巻第四号通巻第四号　昭和五十八年六月七日国鉄首都特別扱承認雑誌第六八九五号